ASTRONOMICAL DATA ANALYSIS SOFTWARE AND SYSTEMS XXXV

COVER ILLUSTRATION:

Cover image description.

Cover image credit.

ASTRONOMICAL SOCIETY OF THE PACIFIC
CONFERENCE SERIES

A SERIES OF BOOKS ON RECENT DEVELOPMENTS IN ASTRONOMY AND ASTROPHYSICS

Volume **VOL Compiled: Tuesday 30th June, 2026@ 01:30**

EDITORIAL STAFF

Managing Editor: Joseph Jensen
Associate Managing Editor: Jonathan Barnes
Publication Manager: Beth Wardell
Publication Consultant: Pepita Ridgeway
e-Book Specialist: Anh Phan

MS 179, Utah Valley University, 800 W. University Parkway, Orem, Utah 84058-5999
Phone: 801-863-8804 E-mail: aspcs@aspbooks.org
E-book site: http://www.aspbooks.org

PUBLICATION COMMITTEE

Jeff Mangum, co-Chair
National Radio Astronomy Observatory

Joseph Jensen, co-Chair
Utah Valley University

Bruce Elmegreen
IBM Watson Research Center

Lynne Hillenbrand
California Institute of Technology

Doug Leonard
San Diego State University

Chris Packham
University of Texas at San Antonio

Amy Mainzer
University of Arizona

The opinions expressed by the authors and editors of ASPCS articles do not necessarily reflect those of the Astronomical Society of the Pacific, the ASP Publication Commitee, or its donors or funding agencies.

ASPCS publications may be found online at http://www.aspbooks.org.

For a complete list of ASPCS Volumes, ASP Monographs, and other ASP publications see http://www.astrosociety.org/pubs.html.

All book order and subscription inquiries should be directed to the ASP at 800-335-2626 (toll-free within the USA) or 415-337-2126, or email service@astrosociety.org

ASTRONOMICAL SOCIETY OF THE PACIFIC
CONFERENCE SERIES

Volume **VOL Compiled: Tuesday 30th June, 2026@ 01:30**

ASTRONOMICAL DATA ANALYSIS SOFTWARE AND SYSTEMS XXXV

Proceedings of a conference held on
online standing in for Victoria, British Columbia, Canada
31 October–4 November 2022

Edited by

Kai Polsterer
HITS gGmbH

David Bohlender
Canadian Astronomy Data Centre, National Research Council Canada, Victoria, British Columbia, Canada

Stephen Gwyn
Canadian Astronomy Data Centre, National Research Council Canada, Victoria, British Columbia, Canada

Adam Hincks
David A. Dunlap Department of Astronomy and Astrophysics, University of Toronto, Toronto, Ontario, Canada

Peter Teuben
Astronomy Department, University of Maryland, College Park, MD, USA

SAN FRANCISCO

ASTRONOMICAL SOCIETY OF THE PACIFIC
390 Ashton Avenue
San Francisco, California, 94112-1722, USA

Phone: 415-337-1100
Fax: 415-337-5205
E-mail: service@astrosociety.org
Web site: www.astrosociety.org
E-books: www.aspbooks.org

First Edition
© **YEAR** by Astronomical Society of the Pacific
ASP Conference Series
All rights reserved.

No part of the material protected by this copyright notice may be reproduced or utilized in any form or by any means—graphic, electronic, or mechanical, including photocopying, taping, recording, or by any information storage and retrieval system—without written permission from the Astronomical Society of the Pacific.

ISBN: **ISBN**
e-book ISBN: **eISBN**

Library of Congress (LOC) Cataloging in Publication (CIP) Data:
Main entry under title
Library of Congress Control Number (LCCN): **LCCN**

Printed in the United States of America by Sheridan Books, Ann Arbor, Michigan.
This book is printed on acid-free paper.

Contents

Preface xiii
S. Gaudet, D. Bohlender, S. Gwyn, A. Hincks and P. Teuben

Participants xvii

Improving Pulse Detection in the X-IFU Readout Chain Using Delayed Derivative Subtraction 1
M. T. Ceballos-Merino, B. Cobo, and N. Cardiel

A new algorithm for the detection and correction of cosmic rays in MEGARA exposures at the Gran Telescopio Canarias 5
N. Cardiel, M. Chillarón-Víctor, S. Pascual, and J. Gallego

Less Cumbersome Units in ADQL 9
M. Demleitner

Targeting YSOs from Large Sky Surveys by Machine Learning 13
Y. Zhang, J. Zhang, and C. Li

Anomaly Detection for Astrometry in the Gaia Mission 17
N. Gaillard and J. L. Hernandez

More than Bytes: Technical and Social Dimensions of Data Lifecycle Management at ESA's Space Science Data Archives 21
R. Bhatawdekar, C. Arviset, B. Merin, G. Marchi, J. Bruijne, D. Baines, A. Masson, M. Bentley, H. Perez, and I. Leon

Generating Astronomical Multiband Images using the Hue-Saturation-Value Colour Space 31
H. Jiang, M. Rahman, J. Scora, and T. Withers

How to Write a Software Paper for the AAS Journals 35
M. Rahman

A Platform to Unify Community-Generated JWST Legacy Data Products 39
J. Scora, M. Rahman, H. Jiang, J. Kemp, K. Iyer, L. Mowla, and T. Withers

Real-time onboard front-end data processing pipelines for the Photospheric Magnetic field Imager on ESA's Vigil mission 43
D. Muraleedharan, J. Hirzberger, J. Staub, H. Michalik, and S. K. Solanki

The FIFI-LS Pipeline at the SOFIA Data Center - New Flux Calibration and Improvements 47
A. Bryant, J. Früh, C. Iserlohe, and B. Knieling

Big data exploration with HiPS3D - a hierarchical visualisation solution for cubic surveys 51
P. Fernique, A. Allen, M. Baumann, T. Boch, C. Bot, and F. Pineau

Euclid Science Archive technical evolution to the Petabyte scale 55
S. Nieto, P. de Teodoro, J. Oliveira, J. Fernandez, C. Rios, J. Prados, V. Olmedo, J. L. Canovas, J. Saiz, V. Fernandez, R. Prieto, E. Ruiz, and P. A. Perez

prosEO: A Cloud-Native Processing Framework for Hybrid Cloud–HPC Astronomy Pipelines . . . 59
F. Raison, T. Bassler, N. C. Meneses, P. Friedl, A. Gidofalvy, and M. Schwinger

Role of General Users in the Lifecycle of Scientific Software . . . 63
B. H. C. Emonts

DS9 FITS Segmentation with Linux and macOS Shell Script Metaprogramming . 73
T. N. Stenborg

CSST Data Processing Pipeline: Architecture & Development . . . 77
B. Zhang

Evolving the Gaia Data Mining Platform to Near-petabyte scale . . . 81
M. Illingworth, B. O'Brien, N. Hambly, S. Harnqvist, and E. Utrilla Molina

PySE's new "sky-eater" mode: source extraction for very large images with over 100,000 sources . . . 85
H. Spreeuw, S. Ali, and T. Millenaar

Kealahou: A Look at the Past, Present, and Future of CFHT's New Queued Service Observing System . . . 91
C. Wipper and C. Usher

Data Formats and Visualisation BoF . . . 95
K. Shortridge

AAOGlimpse goes to Vulkan – and learns some lessons . . . 99
K. Shortridge

Cutana: A High-Performance Tool for Astronomical Image Cutout Generation at Petabyte Scale . . . 103
P. Gómez, L. E. Ruhberg, K., A. Remmelgas, and D. O'Ryan

XMMGPT: Integrating Agentic RAG and Autonomous Agents for Astronomical Data Processing . . . 107
L. Ehrlich, A. Ibarra, R. Saxton, S. Kruk, and M. Doctor

Testing Complex Scientific Software: Key Challenges and Effective Strategies . . 113
S. Castro and A. Wells

Automatic Data Processing with the CASA/ALMA Pipeline . . . 123
D. Muders, K. Berry, S. T. Booth, C.-S. Chang, H. Ezawa, V. C. Geers, L. Gómez, J. E. Hibbard, T. R. Hunter, I. Ihnatiev, R. Indebetouw, T. Izumi, A. A. Kepley, D. Kunneriath, A. Lipnicky, R. A. Loomis, L. T. Maud, T. Nakazato, M. Neeleman, T. Shakunaga, P. Sheehan, K. Shimada, K. Sugimoto, K. Taniguchi, K. Torstensson, A. Vibhute, S. J. Williams, R. Xue, I. Yoon, and A. Yoshino

BlueMUSE: science software for Europe's next large integral field spectrograph . 133
P. M. Weilbacher, C. Vidante, M. Wendt, S. Martens, M. Hayes, A. Adamo, N. Castro, S. Dreizler, A. Jeanneau, A. Kelz, and J. Richard

GammaSky: Edge AI Computing for X- and Gamma-Ray Phenomena in real-time 137
L. Castaldini, A. Bulgarelli, R. Falco, N. Parmiggiani, A. De Rosa, A. Aboudan, L. Amati, P. Calabretto, F. Calzolari, A. Di Piano, V. Fioretti, A. Marinoni, G. Mustafa, G. Panebianco, A. Rizzo, L. Spadoni, and E. Virgilli

Enhancing Access to Legacy Radio Datasets: The ALFALFA Survey in the NRAO Archive 141
B. R. Kent, S. Kagan, J. Sheckard, J. Masters, J. Kern, and M. P. Haynes

An Interactive Metrics Dashboard for the Keck Observatory Archive 145
G, B. Berriman and M. P, Z. Zaw

Automating Data Transfer, Archival Workflows and Data Reduction Pipeline Execution with Apache NiFi 149
J. Coulson, B. Miszalski, S. J. O'Toole, C. Lidman, C. Onken, J. Tocknell, and K. Sheng

Designing the Nightly Digest by ADACS for Rubin Observatory 153
E. Ali, A. Serene, V.R. Becker, S.I. Aranda, and S.J. Kleinman

A New Plotly-Dash-based Query Infrastructure for the Keck Observatory Archive 157
R. Moseley, G. B. Berriman, C. R. Gelino, J. C. Good, J. C. Good, J. C. Good, T. Oluyide, R. Moseley, G. B. Berriman, C. R. Gelino, J. C. Good, M. Lynn, M. Swain, and T. Oluyide

Performance Profiling and Monitoring of the Data Central Ingestion System . . . 161
S. Huynh and J. Tocknell

Adapting Telescope Operations to Science Platforms: How your organisation could benefit from a DevOps Roster 165
J. Tocknell and B. Miszalski

The state of clustered database solutions for astronomical data 169
J. Tocknell

The NADC Dataset Metadata Specification: Enabling FAIR Astronomical Data Discovery 173
Y. Wang, C. Cui, Y. Tao, L. Mi, and B. He

A 30-year journey through science data management 177
C. Arviset, B. Merín, R. Bhatawdekar, D. Baines, M. Bentley, J. de Bruijne, G. de Marchi, A. Masson, V. Navarro, J. Reerink, H. Perez, M. F. Barreiro, J. Cook, P. P. Gil, A. Andres, I. Leon, R. Docasal, H. Norman, and P. Matsson

Localization and Confidence Region Estimation of Short GRBs with the COSI BGO Shield Using a HEALPix-Based Deep Learning Approach 187
N. Parmiggiani, A. Bulgarelli, G. Panebianco, E. Burns, E. Neights, V. Fioretti, I. Martinez-Castellanos, L. Castaldini, A. Ciabattoni, A. Di Piano, R. Falco, S. Gallego, G. Mustafa, P. Patel, A. Rizzo, E. A. Wulf, D. H. Hartmann, C. A. Kierans, J. A. Tomsick, and A. Zoglauer

Development and the Deployment of the Startup System for the ASTRI Mini-Array Project 191
I. Abu, F. Gianotti, A. Bulgarelli, N. Sayes, H. Campos, M. Araya, E. Borciani, P. Bruno, P. Calabretto, L. Castaldini, I. Huerta, V. Conforti, M. Corpora, A. Costa, A. Di Piano, F. Fiordoliva, V. Fioretti, S. Gallozzi, A. Giuliani, A. Grillo, M. Lodi, F. Lucarelli, A. Mandica, G. Mattioli, K. Munari, N. Parmiggiani, G. Pareschi, V. Pastore, F. Russo, S. Scuderi, A. Tacchini, and G. Tosti

Integral Science Legacy Archive: Official Release and Modular Innovation for ESA Science Archives 195
F. Villa, M. Fernández, A. Siles, J. Espinosa, G. Belanger, J. Kajava, and P. Blay

Practical lessons in building collaborative science platforms with open-source tools 199
H. Siejkowski, T. Balawajder, M. Leśniak, J. Kocot, K. Milian, J. Sto, and B. Wenda

The Astrosky Ecosystem: An independent online platform for science communication and social networking 203
E. L. Hunt, V. S. Carpenter, K. W. Cook, D. G. Hilton, M. Asad, J. Jochum, K. Lepo, and J. Zvirzdin

Sustainable data life cycle management for data intensive instruments 207
H. A. Holties and R. P. Pizzo

Refactoring the SIXTE simulator: Towards a more modular code base 211
C. Kirsch, M. Lorenz, L. Dauner, E. G. Gülbahar, N. Reinmann, K. Pal, J. J. R. Stierhof, P. Thalhammer, T. Dauser, and J. Wilms

Accessing and Visualizing Cherenkov Data via an Open Science Web Platform . 215
O. Ates, O. Ates, M. Servillat, C. Boisson, P. Cristofari, B. Lamiré, and R. Savalle

`tilepy`: A Flexible Open-Source Scheduling Engine for Time-Domain and Multi-Messenger Astronomy 219
F. Schussler, H. Ashkar, W. KiendrÃ©bÃ©ogo, M. Seglar-Arroyo, M. de Bony, A. B. E. Ruiz-Velasco, and R. Le Montagner

Astro-COLIBRI: An Innovative Platform for Real-Time Multi-Messenger Astrophysics 223
B. C. Avila, S. Bisero, M. Costa, A. Ciric, I. Jaroschewski, W. Kiendrébéogo, and F. Schussler

Teareduce: a Python package with utilities for teaching reduction techniques in Astronomy 227
N. Cardiel, S. Pascual, M. Chillarón-Víctor, C. Cabello, J. Gallego, J. Zamorano, and M.T. Ceballos

Platform, Participation, and Progress:China's Recent Advances in Astronomical Citizen Science 231
H. Yang, C. Cui, and S. Li

The Vera C. Rubin Observatory Prompt Processing System 237
K. Findeisen, K-T. Lim, D. Speck, H-F. Chiang, E. L. Howard, I. S. Sullivan, and E. C. Bellm

New Visualization and Analysis Capabilities for SDSS DR19 241
M. Taghizadeh-Popp, B. Cherinka, A. Weijmans, J. B. Brownstein, J. Donor, and A. Thakar

Moving objects detection in large catalog database with RANSAC 245
T. Takata, T. Ootsubo, A. Keida, T. Kasuga, J. Firusawa, and H. Furusawa

A Multi-mission strategy for long-term data preservation at the ESAC Science Data Centre 249
M. ArÃ©valo, J. Espinosa, J. P. Cook, Ã. Carasa, I. LeÃ³n, J. Osinde, H. H. PÃ©rez, M. FernÃ¡ndez, B. MerÃn, and R. Bhatawdekar

Multi-messaging: How can uniform access to astronomical archives be achieved? 253
H. Heßling

Representation Learning for Gaia XP DR3 257
B. Doser, K. L. Polsterer, and S. Trujillo-Gomez

Starlink: the 2025A release 261
G. S. Bell, D. S. Berry, M. J. Currie, and S. F. Graves

U(A)I: User-Interfaces and User-Experience in a World Moving towards AI Agents - and even AI Astronomers 265
F. Stoehr, F. Schweder, and K. L. Polsterer

Addressing Big Data Challenges for ESA Space Science Missions 269
S. Kruk, V. Navarro, S. Maksym, P. Gómez, G. Belanger, J. Reerink, C. Arviset, P. Ferruit, V. Pettorino, B. Altieri, J. de Bruijne, F. Marinic, M. López-Caniego, H. Cánovas, and D. Coia

Open SciServer: The (Now Open) Collaborative Science Platform 279
A. Mitschang, G. Lemson, M. Taghizadeh-Popp, and A Thakar

Pipeline of the X-Ray camera of the SVOM mission 283
L. Michel, L. Kleiver, and P. Maggi

CIELOS and the Spanish Virtual Observatory: Exposing astronomical data from Canary Islands Observatories through VO Protocols 287
R. Gutiérrez-Sánchez, E. Solano, J. Falcón-Barroso, C. Allende-Prieto, I. Alonso-Asensio, N. Caon, C. Marín-Galán, J. A. Pérez-Prieto, A. de Vicente, and C. Westendorp

Porting a Multi-Object Spectrograph Mask Design App from IDL to Python . . . 291
E. Jeschke, R. Kackley, M. Tanaka, and M. Kim

Astronomy Open Science Competence Centre in Europe 295
M. Molinaro, M. Allen, J. Wambsganns, E. Solano, B. Cecconi, M. Demleitner, A. Schaaff, H. Heinl, and S. Bertocco

Enhancing the ESA JWST Science Archive Ingestion Pipeline 299
V. FernÃ¡ndez, J. Espinosa, M. ArÃ©valo, M. LÃ³pez-Caniego, D. Baines, A. Labiano, M. FernÃ¡ndez, B. MerÃn, P. Sonnentrucker, and C. Evans

Assembly, Integration, and Verification of the Cherenkov Telescope Array Observatory Data Processing and Preservation System 303
V. Savchenko

PulSKASim: A Pulsar Simulator for SKA-Scale Interferometric Observations . . 307
X. Li and V. Stolyarov

Combining Data and Metadata: Hybrid Tabular Data Formats 311
M. B. Taylor

The ALMA morphological image simlarity search is now online 315
F. Stoehr, A. Farago, S. Curiban, A. Manning, J. Garcia, P.-Y. Hsieh, A. Lipnicky, and A. Plunkett

HiPS2FITS 3D: a new service to generate cube cutouts from HiPS 3D datasets . . 319
T. Boch, M. Baumann, and P. Fernique

Standardizing Software Citation in Astronomy 323
P. Van-Lane, F. Broekgaarden, T. Wagg, J. Duarte, S. Gesing, D. S. Katz, P. Williams, and A. Zonca

On-Orbit Calibration System for the Chinese Space Station Survey Telescope . . 327
J. Ren, C. Liu, Y. Ao, Z. Lin, C. Ling, J. Nie, B. Niu, X. Peng, L. Shao, S. Tan, R. Wang, P. Wei, S. Wei, C. Xu, S. Yao, B. Zhao, and Z. Zhou

Taking a Look through the Incliscope – A Deep Convolutional Mixed Density Network to predict Galactic Inclinations 331
F. Schweder, K. L. Polsterer, and S. Trujillo-Gomez

The SOFIA Data Center (SDC) 337
B. Schulz, A. Bryant, J. Früh, B. Greiner, M. Hütwohl, C. Iserlohe, T. Keilig, B. Knieling, A. Krabbe, K. Schindler, M. Wiedemann, and O. Zeile

25 Years working on ESO Automatic Data Reduction Pipelines: Lessons Learned 341
A. Modigliani

A Robust Snakemake Pipeline for Reducing Complex CII Datasets: The M51 Case Study 345
J. L. Verbena and C. Buchbender

WCSLIB - its origins and development - ADASS XXXV software prize, invited talk 349
M. R. Calabretta

The LISA Science Ground Segment: A Distributed, Multi-Agency Infrastructure for Gravitational Wave Science 359
R. P. de Santayana, N. Lutzgendorf, H. Palacin, A. Basset, F. Wasiak, and T. Littenberg

Simulating antenna voltages for radio observations 363
A. S. Adrian P. Schirra

ScopeSim-Targets: a unified, ESO-compatible target description framework . . . 367
F. Haberhauer, H. Buddelmeijer, and K. Leschinski

Reconstruction of Hubble diagram of Quasars 371
R. Sharma and Y.-S. Song

Re-writing the VizieR catalogue ingestion pipeline 375
I. B. Ivan and G. L. Gilles

An Astronomical Dataset Management Engine for Science Platform 379
J. Han

On-the-fly generated HATS products 383
F. -X. Pineau, P. Fernique, G. Landais, and M. Marchand

Improved quasar and galaxy classifications using the Gaia DR3 387
S. Jamal and C. A. L. Bailer-Jones

BHTOM.space - tool for automatic processing of time-domain data 391
A. Gurgul, Ł. Wyrzykowski, K. Kotysz, J. P. Mikołajczyk, and P. Zieliński

CORA: A Community Platform for Stellar Occultations and Small Solar System Bodies Research 395
M. Kretlow and J.-L. Ortiz

Implementation of HiPS3D in Aladin Lite within the SRCNet framework. 399
M. Baumann, P. Fernique, and T. Boch

Identifying Instruments in the ADS/SciX Corpus with LLM Agents 403
J. Paquin, K. Lockhart, S. Blanco-Cuaresma, A. Accomazzi, M. Kurtz, E. Henneken, F. Grezes, T. Allen, C. Grant, T. Hostetler, M. Templeton, S. Chen, J. Koch, T. Jacovich, D. Chivvis, and F. de Macedo Alves

SIMBAD and Vizier: abuse of service crawling 407
A. Oberto and G Landais

Memory-based Computing: First Results 411
E. Buchholz, H. Heßling, and L. Haupt

Creating an ObsCore builder for data submission 415
N. Barloy and G. Landais

The JMMC Stellar Diameter Catalog version 3, a complete open-source, reproducible & open-data analysis pipeline 419
L. Bourgès, G. Mella, A. Chelli, G. Duvert, P. Cruzalèbes, J.-P. Berger, I. Tallon-Bosc, and D. Mourard

The Julia Ecosystem for Astronomy, Astrophysics, and Geospatial Data Science in 2025 423
M. Kretlow

Maintaining SOFIA Redux 427
B. Greiner, A. Bryant, J. Früh, M. Hütwohl, I. Iserlohe, T. Keilig, B. Knieling, A. Krabbe, K. Schindler, B. Schulz, M. Wiedemann, and O. Zeile

Hybrid Cloud HPC Infrastructure @ DZA 431
B. S. Borra Sagar, B. E. Buchholz Elsa, C. U. Canbolat Utku, R. C. Desing Raja Chithralekha, D. M. Drobek Mark, H. L. Haupt Lars, and H. H. Heßling Hermann

WIVONA - Opening the Virtual Observatory to the Amateur Astronomical Community 435
R. Savalle, J.-P. Godard, C. Cavadore, and D. Valls-Gabaud

transitPipe: a key component of the ESA PLATO Exoplanet Analysis System . . 439
L. C. Smith, S. Hodgkin, F. de Angeli, S. Ahmed, G. Busso, D. Ford, D. Harrison, J. Irwin, D. Murphy, G. Rixon, and N. Walton

Building scientific practices with LLMs in KM3NeT 443
J. Schnabel

pyhysics.tools – Enhancing the Findability of Astronomical Software 447
M. Zacharias, A. Montanari, and S.J. Wagner

Software framework for detecting Pulsar Signals using Digital-Twin and Machine Learning 451
T. Saha and H. Hessling

Small-Scale Science Platform for Active Deep Learning in Large Spectra Archives 455
P. Škoda, O. Burakov, and A. Laiyk

The Universal Bayesian Imaging Kit 459
T. Enßlin, V. Eberle, M. Guardiani, and M. Westerkamp

Author Index 465

Subject Index 471

ASCL Index 476

Preface

Polsterer, K.
Gaudet, S.
Bohlender, D.
Gwyn, S.
Hincks, A.
Teuben, P.

This volume of the Astronomical Society of the Pacific (ASP) Conference Series contains papers that were presented at the 32nd annual conference on Astronomical Data Analysis Software and Systems (ADASS XXXII), held online from October 31 to November 3, 2022.

The ADASS XXXII conference was co-hosted by the Canadian Astronomy Data Centre, part of the Herzberg Astronomy and Astrophysics (HAA) Research Centre of the National Research Council (NRC) of Canada, located in Victoria, British Columbia, Canada, by the Department of Physics and Astronomy, University of Victoria, located in Victoria, British Columbia, Canada, and by the Dunlap Institute for Astronomy and Astrophysics, University of Toronto located in Toronto, Ontario, Canada.

1. Conference Venue

The 2022 edition of the ADASS conference series was initially intended to be an in-person meeting after two years of virtual meetings. The The COVID-19 pandemic was thought to be over, and travel had tentatively re-started. We were looking forward to welcoming colleagues, old and new, to one of the loveliest cities on the planet. However, the Omicron variant put an abrupt stop to this. In January 2022, several large conferences, including the AAS, were canceled at the last minute, with a significant financial loss to both the organizers of the conference and the participants. Some conferences that went ahead with in-person participation experienced COVID-19 outbreaks, leading to the attendees participating online from their hotel rooms. Reluctantly, after considerable thought, the ADASS 2022 organizing committee chose prudence and switched to an online virtual format.

This decision had its drawbacks. After two years of COVID-19, everybody was suffering from Zoom fatigue. Online interactions are less likely to lead to spontaneous conversations that can lead to innovation (that "Aha!" moment). And (let's admit it) in-person meetings are more fun.

That said, the LOC took several steps to use the opportunities presented by an online meeting. The conference fee was kept very low. The agenda was spread as widely as possible around the clock to allow participants not just from North America but also in Europe/Africa and Asia/Australia. While every timezone had its pain point, everybody could participate in some fraction of the sessions. The expanded agenda, the low fee, and the absence of travel meant that several groups who had never attended ADASS, or perhaps only in a very limited capacity, could allow the entire group to participate. Given a taste of what ADASS provides, these groups sent people to ADASS 2023; the pool of ADASS participants has been expanded.

2. Conference Overview

The conference ran from Monday morning (0400 PDT) to Friday evening (1700 PDT). As noted above, the agenda was spread out to include participants from multiple time zones. The agenda was organized predominately by time zone rather than theme. Similarly, these conference proceedings are not ordered by theme but rather by contribution type.

The ten key themes for ADASS XXXII were: 1) Data Processing Frameworks and Algorithms, 2) Longterm Support of (Legacy) Software, 3) Reduction of the environmental footprint of astronomy, 4) Big data, big nightmare?, 5) Astro clouds and science platforms, 6) UI and UX, 7) Robust Systems, 8) Software Architecture and System Design, 9) How to avoid software monoculture while not reinventing the wheel, 10) New techniques and technologies for astronomical software.

There were 46 oral presentations (of which 12 were invited), 82 posters, six focus demonstrations, 7 Birds of a Feather sessions, three tutorials, and four poster lightning sessions.

The conference attracted over 300 participants from all over the world, including developing countries not typically represented at ADASS. As mentioned above, the lack of visa requirements, the convenience of being able to attend from one's couch as well as the significantly less expensive conference registration contributed to the exceptional attendance figure.

The now annual awarding of a Software Prize for "An outstanding contribution to astronomical software" went to the AstroPy Project `https://www.astropy.org/` that has been a community effort with the combination of institutional resources and many dedicated contributors. Erik Tollerud accepted the prize on behalf of the team.

3. Organising Committees and Sponsors

At the time of ADASS XXXII, the Program Organising Committee (POC), chaired by Kathleen Labrie (Gemini), consisted of: Christophe Arviset (ESA-ESAC), Pascal Ballester (ESO), Sebastien Derriere (CDS), Kimberly DuPrie (STScI), Mike Fitzpatrick (NOAO), Stephen Gwyn (CADC), Jorge Ibsen (ALMA), Kathleen Labrie (Gemini), Brian Kent (NRAO), Nuria Lorente (AAO), Jessica Mink (SAO), Fabio Pasian (INAF), Francesco Pierfederici (IRAM), Jose Enrique Ruiz (IAA-CSIC), Rob Seaman (UofA), Yuji Shirasaki (NAOJ), Keith Shortridge (K&V), Oleg Smirnov (UCT-UWC-SKA), Mark Taylor (University of Bristol), Peter Teuben (UMD), and Xiuqin Wu (IPAC).

The Local Organizing Committee (LOC), chaired by Stephen Gwyn (CADC), consisted of Joanna Baylon (UofT), Sarah Bouchard (HAA), Alice Chow (UofT), Lise Fisher (HAA), Séverin Gaudet(CADC), Adam Hincks (UofT), Renée Hloze UofT), Helena Jeeves (CADC), JJ Kavelaars (CADC), Kara Manovich (UofT), Brenda Matthews (HAA), Mark Taylor (UBristol), Karun Thanjavur (UVic), and Jeremy Webb (UofT).

The LOC wishes to thank the fantastic volunteers for their help in making this ADASS a great success.

ADASS XXXII profoundly appreciates the support it received from the host organizations and the sponsorship from the Canada France Hawaii Telescope (CFHT), the European Southern Observatory (ESO), the Space Telescope Science Institute (STScI), the Centre de Données astronomiques de Strasbourg (CDS), and the Infrared Processing and Analysis Center (IPAC).

These proceedings contain **NUMBER**papers representing the invited, oral, and posters presented at the conference, as well as "Birds of a Feather" sessions, focus demos, and tutorials. Publications of astronomical data increasingly make use of the dimension of color. Most of the figures in these proceedings are best viewed in a format that supports color. Readers are therefore encouraged to access the online version, which is available through the ADASS website, `http://www.adass.org` As has become practice now, these proceedings also include an "object index" for the As-

trophysical Source Code Library (ASCL), a free online registry of astronomy software source code of the codes used or mentioned in the papers.

The production of these proceedings has been streamlined thanks mainly to the scripts written by Keith Shortridge and Peter Teuben. We plan to update the proceeding editing workflow code on github[1].

Kai Polsterer
Séverin Gaudet
David Bohlender
Stephen Gwyn
Adam Hincks
Peter Teuben

–The ADASS XXXV Proceedings editors, **MONTH YEAR**.

[1] https://github.com/astroumd/ADASSProceedings

Participants

August Muench, American Astronomical Society, ⟨ august.muench@aas.org ⟩

Alberto Accomazzi, Center for Astrophysics, ⟨ aaccomazzi@cfa.harvard.edu ⟩

Antonio Addis, INAF, ⟨ antonio.addis@inaf.it ⟩

David Aikema, IDIA, University of Cape Town, ⟨ david@idia.ac.za ⟩

Mark Allen, CDS, Observatoire astronomique de Strasbourg, ⟨ mark.allen@astro.unistra.fr ⟩

Alice Allen, Astrophysics Source Code Library/University of Maryland College Park, ⟨ aallen@ascl.net ⟩

Ruben Alvarez, European Space Agency, ⟨ ruben.alvarez@esa.int ⟩

Seyed Amir Hossein Riasati Fard, IOTA-ME, ⟨ amir.h.rst@gmail.com ⟩

Anna Anku, European Space Agency - ESA, ⟨ anna.anku@esa.int ⟩

Maria Arevalo-Sanchez, RHEA Group for ESA, ⟨ maria.arevalo@ext.esa.int ⟩

Keith Arnaud, University of Maryland, ⟨ karnaud@umd.edu ⟩

Christophe Arviset, ESA, ⟨ christophe.arviset@esa.int ⟩

Joseph Asercion, NASA-GSFC, ⟨ joseph.a.asercion@nasa.gov ⟩

Aleksandra Avdeeva, Institute of Astronomy, RAS, ⟨ avdeeva@inasan.ru ⟩

Carlo Baffa, INAF-OAA, ⟨ carlo.baffa@inaf.it ⟩

Deborah Baines, Quasar Science Resources for ESA, ⟨ dbaines@quasarsr.com ⟩

Pascal Ballester, European Southern Observatory, ⟨ pballest@eso.org ⟩

Francisco Javier Ballester Arenas, RHEA group for the European Space Agency, ⟨ javier.ballester@ext.esa.int ⟩

Paul Barrett, The George Washington University, ⟨ pebarrett@gmail.com ⟩

Matthieu Baumann, CDS, Observatoire de Strasbourg, ⟨ matthieu.baumann@astro.unistra.fr ⟩

Joanna Baylon, University of Toronto, ⟨ joanna.baylon@utoronto.ca ⟩

Anja Bekkelien, University of Geneva, ⟨ anja.bekkelien@unige.ch ⟩

Andrea Belfiore, INAF IASF Milan, ⟨ andrea.belfiore@inaf.it ⟩

Graham Bell, East Asian Observatory, ⟨ g.bell@eaobservatory.org ⟩

Joao Bento, Liverpool john moores university, ⟨ j.p.dasilvabento@ljmu.ac.uk ⟩

Aidan Berres, University of Illinois at Urbana-Champaign, ⟨ berres2002@gmail.com ⟩

Bruce Berriman, Caltech/IPAC-NExScI, ⟨ gbb@ipac.caltech.edu ⟩

Kristin Berry, NRAO, ⟨ kberry@nrao.edu ⟩

Sara Bertocco, INAF, ⟨ sara.bertocco@inaf.it ⟩

Sanjay Bhatnagar, National Radio Astronomy Observatory, ⟨ sbhatnag@nrao.edu ⟩

Michael Birchall, AAO Alumni, ⟨ mike.n.birchall@gmail.com ⟩

Susanna Bisogni, INAF-IASF Milano, ⟨ susanna.bisogni@inaf.it ⟩

Thomas Boch, CDS, Observatoire astronomique de Strasbourg,
⟨ thomas.boch@astro.unistra.fr ⟩

David Bohlender, CADC, ⟨ david.bohlender@nrc-cnrc.gc.ca ⟩

Catherine Boisson, LUTH : Observatoire de Paris, ⟨ catherine.boisson@obspm.fr ⟩

Adam Bolton, NSF's NOIRLab, ⟨ adam.bolton@noirlab.edu ⟩

François Bonnarel, CDS, Observatoire astronomique de Strasbourg,
⟨ francois.bonnarel@astro.unistra.fr ⟩

Lucian Botha, SAAO, ⟨ lucian@saao.ac.za ⟩

Sarah Bouchard, Herzberg Astronomy and Astrophysics,
⟨ sarah.bouchard@nrc-cnrc.gc.ca ⟩

Max Brodheim, W. M. Keck Observatory, ⟨ mbrodheim@keck.hawaii.edu ⟩

Matthew Brown, W. M. Keck Observatory, ⟨ mbrown@keck.hawaii.edu ⟩

Nicholas Bruce, National Research Council Canada,
⟨ nicholas.bruce@nrc-cnrc.gc.ca ⟩

Anne Buckner, University of Exeter, ⟨ a.buckner@exeter.ac.uk ⟩

Hugo Buddelmeijer, Kapteyn Institute University of Groningen,
⟨ hugo@buddelmeijer.nl ⟩

Wybren Buijs, Jive, ⟨ buijs@jive.eu ⟩

Andrea Bulgarelli, INAF/OAS Bologna, ⟨ andrea.bulgarelli@inaf.it ⟩

Suela Buzi, Caltech/IPAC, ⟨ suela@ipac.caltech.edu ⟩

Giorgio Calderone, INAF, ⟨ giorgio.calderone@inaf.it ⟩

Lachlan Campbell, Pawsey Supercomputing Centre,
⟨ lachlan.campbell@pawsey.org.au ⟩

Jonathan Carifio, Center for Astrophysics | Harvard & Smithsonian,
⟨ jonathan.carifio@cfa.harvard.edu ⟩

Benoit Carry, CNRS, Lagrange Lab., ⟨ benoit.carry@oca.eu ⟩

Helena Castro, European Space Agency, ⟨ helena.vicente.de.castro@esa.int ⟩

Sandra Castro, ESO, ⟨ scastro@eso.org ⟩

Christine Chang, Caltech/IPAC, ⟨ cschang@ipac.caltech.edu ⟩

Malynda Chizek Frouard, Space Telescope Science Institute,
⟨ malyndacf@gmail.com ⟩

Liz Chock, W. M. Keck Observatory, ⟨ echock@keck.hawaii.edu ⟩

Tamara Civera, CEFCA, ⟨ tcivera@cefca.es ⟩

Andrew Connolly, University of Washington, ⟨ ajc@astro.washington.edu ⟩

Nick Cox, ACRI-ST, ⟨ nick.cox@acri-st.fr ⟩

André Csillaghy, FHNW Insitute for Data Science, ⟨ andre.csillaghy@fhnw.ch ⟩

Guido Cupani, OATS INAF, ⟨ guido.cupani@inaf.it ⟩

Keto D. Zhang, Caltech/IPAC, ⟨ keto.zhang@gmail.com ⟩

Ashkbiz Danehkar, Eureka Scientific, ⟨ ashkbiz.danehkar@gmail.com ⟩

Arpan Das, EPFL (École polytechnique fédérale de Lausanne), ⟨ arpan.das@epfl.ch ⟩

Jan David Mol, Stichting ASTRON, ⟨ mol@astron.nl ⟩

Elizabeth Davies, Australian Astronomical Optics - Macquarie University, ⟨ elizabeth.mannering@mq.edu.au ⟩

Jonathan Davis, Univerisity of Utah, ⟨ jonathandaviddavis@gmail.com ⟩

Pilar de Teodoro, ESA, ⟨ pilar.teodoro@ext.esa.int ⟩

Miguel del Prado (ESO), ESO, ⟨ miguel.delpradoaranda@eso.org ⟩

David A. Del Rizzo, National Research Council Canada, ⟨ dave.delrizzo@nrc-cnrc.gc.ca ⟩

Melissa Delucchi, LINCC CMU, ⟨ delucchi@andrew.cmu.edu ⟩

James Dempsey, CSIRO Australia, ⟨ james.dempsey@csiro.au ⟩

Vladimir Dergachev, Albert Einstein Institute Hannover, ⟨ vladimir.dergachev@aei.mpg.de ⟩

Sébastien Derriere, Centre de Données astronomiques de Strasbourg, ⟨ sebastien.derriere@astro.unistra.fr ⟩

Erik Deul, Leiden Observatory, ⟨ deul@strw.leidenuniv.nl ⟩

Miguel Doctor Yuste, TLZ for ESA, ⟨ miguel.doctor@ext.esa.int ⟩

Axel Donath, CFA, ⟨ axel.donath@cfa.harvard.edu ⟩

Jayce Dowell, University of New Mexico, ⟨ jdowell@unm.edu ⟩

Trisha Doyle, Innovim/NASA Goddard Space Flight Center, ⟨ trisha.f.doyle@nasa.gov ⟩

Gregory Dubois-Felsmann, Caltech/IPAC, ⟨ gpdf@ipac.caltech.edu ⟩

Kimberly DuPrie, Space Telescope Science Institute, ⟨ kduprie@stsci.edu ⟩

Daniel Durand, CDS and CADC, ⟨ durand.daniel@gmail.com ⟩

Dakota Dutko, NASA, ⟨ dakota.c.dutko@nasa.gov ⟩

Heinz-Bernd Eggenstein, Max Planck Institute for Gravitational Physics/Leibniz Universität Hannover, ⟨ heinz-bernd.eggenstein@aei.mpg.de ⟩

Satoshi Eguchi, Department of Applied Physics, Faculty of Science, Fukuoka University, ⟨ satoshieguchi@fukuoka-u.ac.jp ⟩

Bjorn Emonts, NRAO Charlottesville, ⟨ bjornemonts@gmail.com ⟩

Neil Ernst, University of Victoria, ⟨ nernst@uvic.ca ⟩

Javier Espinosa Aranda, ESA, ⟨ javier.espinosa@ext.esa.int ⟩

Paolo Estrada, University of Belize, ⟨ paolo.estrada.m@gmail.com ⟩

Janet Evans, Center for Astrophysics | Harvard & Smithsonian, ⟨ janet@cfa.harvard.edu ⟩

Sébastien Fabbro, NRC-HAA CADC, ⟨ sébastien.fabbro@nrc-cnrc.gc.ca ⟩

Humberto Farias, Universidad Técnica Federico Santa María / Universidad de La Serena, ⟨ humberto.farias@usm.cl ⟩

Tony Farrell, AAO Macquarie , ⟨ tony.farrell@mq.edu.au ⟩

Carlos Feinstein, IALP-Conicet, ⟨ carlosfb@gmail.com ⟩

Monica Fernandez Barreiro, Monica Fernandez Barreiro, RHEA Group for ESA, ⟨ monica.fernandez@sciops.esa.int ⟩

Nuwanthika Fernando, Australian Astronomical Optics - Macquarie University, ⟨ nuwanthika.fernando@mq.edu.au ⟩

Pierre Fernique, CDS, Observatoire astronomique de Strasbourg, ⟨ pierre.fernique@astro.unistra.fr ⟩

Lise Fisher, Herzberg Astronomy and Astrophysics, ⟨ lise.fisher@nrc-cnrc.gc.ca ⟩

Mike Fitzpatrick, NSF's NOIRLab, ⟨ mike.fitzpatrick@noirlab.edu ⟩

Alicia Flint, CDS, Observatoire astronomique de Strasbourg, ⟨ alicia.vanhulle@astro.unistra.fr ⟩

Miguel Flores, University of Guadalajara, ⟨ miguel.flores.rod@gmail.com ⟩

Francisco Forster, Universidad de Chile, ⟨ francisco.forster@gmail.com ⟩

Thierry Forveille, Astronomy & Astrophysics, ⟨ thierry.forveille@univ-grenoble-alpes.fr ⟩

Paolo Franzetti, INAF, ⟨ paolo.franzetti@inaf.it ⟩

Wesley Fraser, NRC-HAA, ⟨ wesley.fraser@nrc-cnrc.gc.ca ⟩

Lucas Fuhrman, W. M. Keck Observatory, ⟨ lfuhrman@keck.hawaii.edu ⟩

Marco Fumana, INAF, ⟨ marco.fumana@inaf.it ⟩

Hisanori Furusawa, National Astronomical Observatory of Japan, ⟨ furusawa.hisanori@nao.ac.jp ⟩

Junko Furusawa, National Astronomical Observatory of Japan, ⟨ furusawa.junko@nao.ac.jp ⟩

Bryan Gaensler, Dunlap Institute for Astronomy and Astrophysics, ⟨ bryan.gaensler@utoronto.ca ⟩

Anastasia Galkin, Leibniz Institute for Astrophysics Potsdam (AIP), ⟨ agalkin@aip.de ⟩

Vincenzo Galluzzi, INAF - Istituto di Radioastronomia, ⟨ vincenzo.galluzzi@inaf.it ⟩

Enrique Garcia, CERN, ⟨ enrique.garcia.garcia@cern.ch ⟩

Séverin Gaudet, NRC-HAA, ⟨ severin.gaudet@nrc-cnrc.gc.ca ⟩

Gordon German, Commonwealth Scientific and Industrial Research Organisation, ⟨ gwhgerman@gmail.com ⟩

Meredith Gibb, NASA/GSFC, ⟨ meredith.gibb@nasa.gov ⟩

Kim Gillies, Thirty Meter Telescope International Observatory, ⟨ kgillies@tmt.org ⟩

Hunter Goddard, Kansas State University, ⟨ hbgoddard@ksu.edu ⟩

Sharon Goliath, CADC, ⟨ sharon.goliath@nrc-cnrc.gc.ca ⟩

Michelle Gower, NCSA/University of Illinois Urbana-Champaign, ⟨ mgower@illinois.edu ⟩

Yan Grange, ASTRON, ⟨ grange@astron.nl ⟩

Felix Grezes, NASA/ADS, ⟨ felix.grezes@cfa.harvard.edu ⟩

Steve Groom, Caltech/IPAC, ⟨ sgroom@ipac.caltech.edu ⟩

Julia Gschwend, LIneA, ⟨ julia@linea.org.br ⟩

Stephen Gwyn, Canadian Astronomy Data Centre, ⟨ stephen.gwyn@nrc-cnrc.gc.ca ⟩

Jonas Haase, MPE, ⟨ jhaase@mpe.mpg.de ⟩

Patricia Hall, NASA/Innovim, ⟨ patkobak@gmail.com ⟩

Daniel Hall, Johns Hopkins University, ⟨ dhall9734@gmail.com ⟩

Miranda Hawarden-Ogata, NASA IRTF; Institute for Astronomy; UH-Manoa, ⟨ hawarden@hawaii.edu ⟩

Hendrik Heinl, Centre de Donees de Strasbourg, ⟨ hendrik.heinl@astro.unistra.fr ⟩

Lorena Hernandez, Millennium Institute of Astrophysics, ⟨ lorena.hernandez@uv.cl ⟩

David Herrera, AURA/NOIRLab, ⟨ david.herrera@noirlab.edu ⟩

Christian Hettlage, SAAO/SALT, ⟨ c.hettlage@saao.nrf.ac.za ⟩

Robert Hill, ADNET Systems, Inc., ⟨ robert.s.hill@nasa.gov ⟩

Adam Hincks, Dunlap Institute and Department of Astronomy and Astrophysics, University of Toronto, ⟨ adam.hincks@utoronto.ca ⟩

Renee Hlozek, Dunlap Institute and Department of Astronomy and Astrophysics, University of Toronto, ⟨ hlozek@dunlap.utoronto.ca ⟩

Renee Hlozek, Dunlap Institute, University of Toronto, ⟨ renee.hlozek@gmail.com ⟩

Paul Huwe, Space Telescope Science Institute (STScI), ⟨ phuwe@stsci.edu ⟩

Minh Huynh, CSIRO, ⟨ minh.huynh@csiro.au ⟩

Tess Jaffe, NASA HEASARC, ⟨ tess.jaffe@nasa.gov ⟩

Tammo Jan Dijkema, ASTRON, ⟨ dijkema@astron.nl ⟩

Helena Jeeves, NRC-HAA CADC, ⟨ helena.jeeves@nrc-cnrc.gc.ca ⟩

Tim Jenness, AURA/Rubin Observatory, ⟨ tjenness@lsst.org ⟩

Moloko Jeremia Hlakola, South African Astronomical Observatory, ⟨ molokoh@saao.ac.za ⟩

Eric Jeschke, Subaru Telescope, National Astronomical Observatory of Japan, ⟨ eric@naoj.org ⟩

Emmanuel Joliet, Caltech/IPAC, ⟨ ejoliet@ipac.caltech.edu ⟩

Carina Kan, NASA / HEASARC GSFC, ⟨ carina.d.kan@nasa.gov ⟩

JJ Kavelaars, CADC, ⟨ jj.kavelaars@nrc-cnrc.gc.ca ⟩

Michael Keane, Space Telescope Science Institute, ⟨ mkeane@stsci.edu ⟩

Thad Kenny, UK Astronomy Technology Centre, ⟨ thaddeus.kenny@stfc.ac.uk ⟩

Brian Kent, National Radio Astronomy Observatory, ⟨ bkent@nrao.edu ⟩

Miriam Keppler, University of Arizona, ⟨ keppler@arizona.edu ⟩

Kathleen Kiker, B612 Foundation, ⟨ kathleen@b612foundation.org ⟩

Kechil Kirkham, IDIA, University of Cape Town, ⟨ kechil@idia.ac.za ⟩

Klaas Kliffen, ASTRON, ⟨ kliffen@astron.nl ⟩

Adriaan Klinkenberg, Elsevier, ⟨ f.klinkenberg@elsevier.com ⟩

Jürgen Knödlseder, IRAP, ⟨ jknodlseder@irap.omp.eu ⟩

Yuzuki Koga, University of Hiroshima, ⟨ koga@astro.hiroshima-u.ac.jp ⟩

Paula Kornecki, LUTH ; Observatoire de Paris, ⟨ paula.kornecki@obspm.fr ⟩

Martin Kuemmel, LMU Munich, ⟨ mkuemmel@usm.lmu.de ⟩

Kathleen Labrie, NOIRLab/Gemini Observatory, ⟨ kathleen.labrie@noirlab.edu ⟩

Dustin Lagoy, National Research Council Canada, ⟨ dustin.lagoy@nrc-cnrc.gc.ca ⟩

Anastasia Laity, Caltech/IPAC, ⟨ laity@ipac.caltech.edu ⟩

Gilles Landais, CDS, Observatoire astronomique de Strasbourg, ⟨ gilles.landais@astro.unistra.fr ⟩

Fanna Lautenbach, ASTRON, ⟨ lautenbach@astron.nl ⟩

Casey Law, Caltech, ⟨ caseylaw@caltech.edu ⟩

David Law, Liverpool John Moores University, ⟨ d.j.law@2021.ljmu.ac.uk ⟩

Jae-Joon Lee, Korea Astronomy and Space Science Institute, ⟨ lee.j.joon@gmail.com ⟩

Chien-Hsiu Lee, W. M. Keck Observatory, ⟨ clee@keck.hawaii.edu ⟩

Florian Leitgeb, Deutsches Elektronen-Synchrotron DESY, ⟨ florian.leitgeb@desy.de ⟩

Frederic Lemmel, ESA, ⟨ frederic.lemmel@esa.int ⟩

Kieran Leschinski, University of Vienna, ⟨ kieran.leschinski@univie.ac.at ⟩

Ping Lin, University of Toronto, ⟨ ping.lin@utoronto.ca ⟩

Tak Lo, IPAC/Caltech, ⟨ tlo@ipac.caltech.edu ⟩

Min Long, Boise State University, ⟨ minlong@boisestate.edu ⟩

Marcel Loose, ASTRON, ⟨ loose@astron.nl ⟩

Marcos Lopez-Caniego, Aurora Technology for ESA, ⟨ marcos.lopez.caniego.alcarria@ext.esa.int ⟩

Nuria Lorente, AAO - Australian Astronomical Optics, Macquarie University, ⟨ nuria.lorente@mq.edu.au ⟩

Mireille Louys, CDS , Observatoire Astronomique & Icube Laboratory, University Strasbourg, ⟨ mireille.louys@unistra.fr ⟩

Nate Lust, Princeton University / Rubin Observatory, ⟨ nlust@astro.princeton.edu ⟩

Nhlavutelo Macebele, SAAO, ⟨ ne.macebele@saao.nrf.ac.za ⟩

Jeff Mader, W. M. Keck Observatory, ⟨ jmader@keck.hawaii.edu ⟩

Gernot Maier, DESY, ⟨ gernot.maier@desy.de ⟩

Brian Major, Canadian Astronomy Data Centre, ⟨ brian.major@nrc-cnrc.gc.ca ⟩

Kirill Makan, Leibniz Institute for Astrophysics Potsdam (AIP), ⟨ kmakan@aip.de ⟩

Mattia Mancini, ASTRON, ⟨ mancini@astron.nl ⟩

Jeff Mangum, NRAO, ⟨ jmangum@nrao.edu ⟩

Alisdair Manning, ESO Garching, ⟨ amanning@eso.org ⟩

Manon Marchand, Observatoire Astronomique de Strasbourg, ⟨ manon.marchand@astro.unistra.fr ⟩

Filip Marinic, European Space Agency, ⟨ filip.marinic@esa.int ⟩

Gianluca Marotta, INAF - OAA, ⟨ gianluca.marotta@inaf.it ⟩

Anthony Marston, European Space Agency, ⟨ tmarston@sciops.esa.int ⟩

Yagi Masafumi, National Astronomical Observatory of Japan, ⟨ yagi.masafumi@nao.ac.jp ⟩

Brenda Matthews, NRC-HAA, ⟨ bcmatthews.herzberg@gmail.com ⟩

Douglas McElroy, IPAC, California Institute of Technology, ⟨ dlm@ipac.caltech.edu ⟩

Brian McLean, Space Telescope Science Institute, ⟨ mclean@stsci.edu ⟩

David Melkumyan, Deutsches Elektronen-Synchrotron, ⟨ david.melkumyan@desy.de ⟩

Bruno Merín, ESA, ⟨ bruno.merin@esa.int ⟩

Jessica Mink, Smithsonian Astrophysical Observatory, ⟨ jmink@cfa.harvard.edu ⟩

Brent Miszalski, Australian Astronomical Optics – Macquarie, ⟨ brent.miszalski@mq.edu.au ⟩

Chaka Mofokeng, SAAO/SALT, ⟨ c.mofokeng@saao.nrf.ac.za ⟩

Marco Molinaro, INAF, ⟨ marco.molinaro@inaf.it ⟩

Kgomotso Monnakgotla, NRF/SARAO, ⟨ kmonnakgotla@sarao.ac.za ⟩

Dave Morris, University of Edinburgh, ⟨ dmr@roe.ac.uk ⟩

Ahmed Mubashir Khan, ESO Garching, ⟨ akhan@eso.org ⟩

Fritz Mueller, SLAC National Accelerator Laboratory, ⟨ fritzm@slac.stanford.edu ⟩

Karl Mueller, SLAC National Accelerator Laboratory, ⟨ fritzm@fritzm.org ⟩

Oindabi Mukherjee, Michigan Technological University, ⟨ omukherj@mtu.edu ⟩

Alejandra Muñoz Arancibia, Millennium Institute of Astrophysics, ⟨ amma.19@gmail.com ⟩

Yusuke Nakakoji, Shibaura Institute of Technology, ⟨ ynakakoji731@gmail.com ⟩

Vicente Navarro, European Space Agency - ESA, ⟨ vicente.navarro@esa.int ⟩

Fausto Navarro, The Johns Hopkins University, ⟨ fnavarr3@jhu.edu ⟩

Xola Ndaliso, SAAO/SALT, ⟨ xola@saao.ac.za ⟩

Jon Nielsen, The Australian National University, ⟨ jon.nielsen@anu.edu.au ⟩

Sara Nieto Rodriguez, RHEA Systems, ⟨ sara.nieto@esa.int ⟩

Robert Nikutta, NOIRLab / Astro Data Lab, ⟨ robert.nikutta@noirlab.edu ⟩

Ricky Nilsson, Caltech/IPAC, ⟨ ricky@ipac.caltech.edu ⟩

Mike Nolta, CITA, University of Toronto, ⟨ nolta@scinet.utoronto.ca ⟩

John O'Meara, W. M. Keck Observatory, ⟨ jomeara@keck.hawaii.edu ⟩

William O'Mullane, Vera C. Rubin Observatory, ⟨ womullan@lsst.org ⟩

Simon O'Toole, Australian Astronomical Optics - Macquarie University, ⟨ simon.otoole@mq.edu.au ⟩

Anaïs Oberto, CDS, Observatoire astronomique de Strasbourg, ⟨ anais.oberto@astro.unistra.fr ⟩

Pierre Ocvirk, CDS, Observatoire astronomique de Strasbourg, ⟨ pierre.ocvirk@astro.unistra.fr ⟩

Takafumi Ootsubo, National Astronomical Observatory of Japan, ⟨ takafumi.ootsubo@nao.ac.jp ⟩

Ralf Palsa, ESO Garching, ⟨ rpalsa@eso.org ⟩

Nicolò Parmiggiani, INAF/OAS Bologna, ⟨ nicolo.parmiggiani@inaf.it ⟩

Manuel Parra-Royón, Institute of Astrophysics of Andalusia - CSIC, ⟨ mparra@iaa.es ⟩

Fabio Pasian, INAF-OATs (Trieste), ⟨ fabio.pasian@inaf.it ⟩

Kalyani Pedamkar, INAF- Osservatorio Astronomico di Trieste, ⟨ kalyani.pedamkar@inaf.it ⟩

Efrain Perez-Solis, ADNET Systems Inc., ⟨ efrain.c.perez-solis@nasa.gov ⟩

Francois-Xavier Pineau, Université de Strasbourg, ⟨ francois-xavier.pineau@astro.unistra.fr ⟩

Adrianna Pińska, IDIA, University of Cape Town, ⟨ adrianna@idia.ac.za ⟩

Stanislaw Podgorski, ESO Garching, ⟨ stanislaw.podgorski@eso.org ⟩

Ondřej Podsztavek, Faculty of Information Technology, Czech Technical University in Prague, ⟨ podszond@fit.cvut.cz ⟩

Kai Polsterer, HITS gGmbH, ⟨ kai.polsterer@h-its.org ⟩

Kartikeya Puri, Caltech/IPAC, ⟨ kapuri@ipac.caltech.edu ⟩

Judith Racusin, NASA/GSFC, ⟨ judith.racusin@nasa.gov ⟩

Troy Raen, Caltech/IPAC, ⟨ raen@ipac.caltech.edu ⟩

Venkatessh Ramakrishnan, Finnish Centre for Astronomy with ESO, ⟨ venkatessh.ramakrishnan@aalto.fi ⟩

Suhasini S Rao, University of Alberta,

Manuela Rauch, Know Center, ⟨ mrauch@know-center.at ⟩

Anne Raugh, Univeristy of Maryland - Astronomy, ⟨ araugh@umd.edu ⟩

Jan Reerink, ESA, ⟨ jan.reerink@esa.int ⟩

Rosly Renil, NRF/SARAO, ⟨ rosly@sarao.ac.za ⟩

Alexander Reustle, NASA/GSFC - FERMI, ⟨ alexreustle@gmail.com ⟩

Samira Rezaei, Drexel University College of Engineering, ⟨ rezaei@strw.leidenuniv.nl ⟩

Luca Rizzi, National Science Foundation, ⟨ lrizzi@nsf.gov ⟩

David Rodriguez, Space Telescope Science Institute, ⟨ drodriguez@stsci.edu ⟩

Sergey Rostopchin, UT-McDonald Observatory-Hobby Eberly Telescope, ⟨ rhenry@astro.as.utexas.edu ⟩

Kristof Rozgonyi, Ludwig Maximilian University of Munich, ⟨ kristof.rozgonyi@physik.uni-muenchen.de ⟩

Ben Rusholme, Caltech/IPAC, ⟨ rusholme@ipac.caltech.edu ⟩

Felix Stoehr, ESO / ALMA, ⟨ fstoehr@eso.org ⟩

Mike Sample, Sample Inc, ⟨ msample.adass_202208@sample.com ⟩

Suhasini Satish Rao, University of Alberta, ⟨ suhasin1@ualberta.ca ⟩

Destry Saul, NoirLab, ⟨ destry.saul@noirlab.edu ⟩

Renaud Savalle, PADC, Paris Observatory, ⟨ renaud.savalle@obspm.fr ⟩

Reinhold Schaaf, Universität Bonn, ⟨ rschaaf@astro.uni-bonn.de ⟩

André Schaaff, CDS, Observatoire astronomique de Strasbourg, ⟨ andre.schaaff@astro.unistra.fr ⟩

Matthias Schartner, ETH Zurich, ⟨ mschartner@ethz.ch ⟩

Jutta Schnabel, Friedrich Alexander University, ⟨ jutta.schnabel@fau.de ⟩

Ullrich Schwanke, Humboldt University, Berlin, ⟨ schwanke@physik.hu-berlin.de ⟩

Eva Sciacca, INAF Osservatorio Astrofisico di Catania, ⟨ eva.sciacca@inaf.it ⟩

Victoria Sciaroni, Space Telescope Science Institute, ⟨ vsciaroni@stsci.edu ⟩

Rob Seaman, University Of Arizona, ⟨ rseaman@arizona.edu ⟩

Constanze Seibert, ESA, ⟨ constanze.seibert@esa.int ⟩

Mathieu Servillat, Observatoire de Paris, ⟨ mathieu.servillat@obspm.fr ⟩

Konstantinos Servis, CSIRO - Pawsey Supercomputing Research Centre, ⟨ knservis@gmail.com ⟩

Lior Shamir, Kansas State University, ⟨ lshamir@mtu.edu ⟩

Austin Shen, CSIRO, ⟨ austin.shen@csiro.au ⟩

Sharon Shen, Space telescope science institute, ⟨ sshen@stsci.edu ⟩

Kate Sheng, Australian Astronomical Optics - Macquarie University, ⟨ kate.sheng@mq.edu.au ⟩

Yuji Shirasaki, NAOJ, ⟨ yuji.shirasaki@nao.ac.jp ⟩

Makoto Shizugami, National Astronomical Observatory of Japan, ⟨ m.shizugami@nao.ac.jp ⟩

Keith Shortridge, K&V, ⟨ keith@knaveandvarlet.com.au ⟩

Sergey Shumko, NJIT/Big Bear Solar Observatory, ⟨ shoumko@njit.edu ⟩

David Shupe, Caltech/IPAC, ⟨ shupe@ipac.caltech.edu ⟩

Judith Silverman, Caltech/IPAC, ⟨ judith@ipac.caltech.edu ⟩

Brigitta Sipőcz, Caltech/IPAC, ⟨ bsipocz@ipac.caltech.edu ⟩

Alexander Sivitilli, University of Cape Town, ⟨ alexandersivitilli@gmail.com ⟩

Oleg Smirnov, Rhodes University, ⟨ o.smirnov@ru.ac.za ⟩

Benedikt Soja, ETH Zurich, ⟨ soja@ethz.ch ⟩

Renato Sortino, Istituto Nazionale di Astrofisica, ⟨ renato.sortino@inaf.it ⟩

Christian Soto, Space Telescope Science Institute, ⟨csoto@stsci.edu⟩

Johannes Spreeuw, The Netherlands eScience Center, ⟨h.spreeuw@esciencecenter.nl⟩

Travis Stenborg, University of Sydney, ⟨travis.stenborg@sydney.edu.au⟩

Jeff Stoner, Southwest Research Institute, ⟨jeff.stoner@swri.org⟩

Ole Streicher, Leibniz Institute for Astrophysics, ⟨ole@aip.de⟩

Mitchell Stringfellow, CSIRO Australia, ⟨mitchell.stringfellow@csiro.au⟩

Igor Suarez-Sola, NOIRLab/CSDC/Datalab, ⟨igor.suarez-sola@noirlab.edu⟩

Yashil Sukurdeep, The Johns Hopkins University, ⟨yashil.sukurdeep@jhu.edu⟩

John Swinbank, ASTRON, ⟨swinbank@astron.nl⟩

Giuliano Taffoni, INAF - OATS, ⟨giuliano.taffoni@inaf.it⟩

Satoshi Takashima, University of Tokyo, ⟨satoshi.takashima@phys.s.u-tokyo.ac.jp⟩

Mark Taylor, University of Bristol, ⟨m.b.taylor@bristol.ac.uk⟩

Russ Taylor, University of the Western Cape, ⟨russ@idia.ac.za⟩

Hossen Teimoorinia, CADC, ⟨hossteim@uvic.ca⟩

Nathaniel Tellis, B612 Foundation, ⟨nate@b612foundation.org⟩

Peter Tenenbaum, SETI Institute / NASA Ames Research Center, ⟨peter.tenenbaum@nasa.gov⟩

Jonathan Tennyson, RAS Techniques and Instruments, ⟨j.tennyson@ucl.ac.uk⟩

Mairan Teodoro, STScI, ⟨mteodoro@stsci.edu⟩

Peter J. Teuben, University of Maryland, ⟨teuben@astro.umd.edu⟩

Karun Thanjavur, UVic, ⟨karun@uvic.ca⟩

Deepashri Thatte, Space Telescope Science Institute, ⟨thatte@stsci.edu⟩

Sebastian Thiel, Max-Planck Institute for Radioastronomy, ⟨sthiel@mpifr-bonn.mpg.de⟩

James Tocknell, AAO, ⟨james.tocknell@mq.edu.au⟩

Erik Tollerud, Space Telescope Science Institute, ⟨erik.tollerud@gmail.com⟩

Tyler Tucker, W. M. Keck Observatory, ⟨ttucker@keck.hawaii.edu⟩

Makoto Uemura, Hiroshima University, ⟨uemuram@hiroshima-u.ac.jp⟩

Francisco Valdes, NOIRLab, ⟨frank.valdes@noirlab.edu⟩

Ilse van Bemmel, JIVE, ⟨bemmel@jive.eu⟩

Wim Van Cappellen, ASTRON, ⟨facturen@astron.nl⟩

Gijs Verdoes Kleijn, University of Gronigen, ⟨g.a.verdoes.kleijn@rug.nl⟩

Nico Vermaas, ASTRON, ⟨vermaas@astron.nl⟩

Jose Vicente Perea-Calderon, RHEA Systems, ⟨jose.vicente.perea.calderon@ext.esa.int⟩

WJ Vriend, Rijks Universiteit Groningen, ⟨wjvriend@astro.rug.nl⟩

Thomas Vuillaume, IN2P3, ⟨thomas.vuillaume@lapp.in2p3.fr⟩

Shin-Ywan Wang, Caltech/IPAC, ⟨ cwang@ipac.caltech.edu ⟩

Jeremy Webb, University of Toronto, ⟨ jeremy.webb@utoronto.ca ⟩

Jason Weiss, TMT, ⟨ jweiss@tmt.org ⟩

Jack White, University of Oxford, ⟨ jackajwhite@icloud.com ⟩

Andreas Wicenec, ICRAR, ⟨ andreas.wicenec@icrar.org ⟩

Peter Williams, CfA|H&S / AAS, ⟨ pwilliams@cfa.harvard.edu ⟩

Chris Willott, CADC, ⟨ christopher.willott@nrc-cnrc.gc.ca ⟩

Avery Wold, Caltech/IPAC, ⟨ awold@ipac.caltech.edu ⟩

Xiuqin Wu, Infrared Processing & Analysis Center, ⟨ xiuqin@ipac.caltech.edu ⟩

Rui Xue, National Radio Astronomy Observatory, ⟨ rxue@nrao.edu ⟩

Suk Yee Yong, Macquarie University, ⟨ sukyee.yong@mq.edu.au ⟩

Jinmi Yoon, Space Telescope Science Institute, ⟨ jyoon@stsci.edu ⟩

Kenji Yoshida, Shibaura Institute of Technology, ⟨ yoshida@shibaura-it.ac.jp ⟩

Adam Zadrozny, National Centre for Nuclear Research,
⟨ adam.zadrozny@gmail.com ⟩

Stefano Zampieri, ESO, ⟨ szampier@eso.org ⟩

Alessandra Zanichelli, INAF - Istituto di Radioastronomia,
⟨ alessandra.zanichelli@inaf.it ⟩

Angela Zhang, CalTech, ⟨ azhang@ipac.caltech.edu ⟩

Abdu Zoghbi, CRESST, University of Maryland CP. NASA-GSFC,
⟨ abderahmen.zoghbi@nasa.gov ⟩

Petr Škoda, Astronomical Institute CAS, ⟨ skoda@sunstel.asu.cas.cz ⟩

Astronomical Data Analysis Software and Systems XXXV
*ASP Conference Series, Vol. **VOL** Compiled: Tuesday 30th June, 2026@ 01:30*
K. Polsterer, D. Bohlender, S. Gwyn, A. Hincks and P. Teuben, eds.
*© **YEAR** Astronomical Society of the Pacific*

ZZZDEBUG!003.tex
Ceballos-Merino, M. T.|tex
Cobo, B.
Cardiel, N.
algorithm
data!analysis
instruments!individual!X-I
astronomy!pulse
detection
techniques!event
detection
observatories!space-based!
SIRENA! ascl:2307.013

Improving Pulse Detection in the X-IFU Readout Chain Using Delayed Derivative Subtraction

M. Teresa Ceballos-Merino,[1] Beatriz Cobo,[1] and Nicolás Cardiel[2,3]

[1]*Instituto de Física de Cantabria, CSIC-Universidad de Cantabria, Santander, Cantabria, Spain;* ceballos@ifca.unican.es

[2]*Departamento de Física de la Tierra y Astrofísica, Facultad CC. Físicas, Universidad Complutense de Madrid, Madrid, Spain*

[3]*Instituto de Física de Partículas y del Cosmos, IPARCOS, Facultad CC. Físicas, Universidad Complutense de Madrid, Madrid, Spain*

Abstract. The X-ray Integral Field Unit (X-IFU) onboard the Athena observatory employs Transition Edge Sensor (TES) microcalorimeters to achieve high-resolution spectroscopy. Accurate triggering of photon-induced pulses in the on-board Event Processor is essential for reliable event reconstruction, particularly under high count-rate conditions where pulse overlap becomes significant. Standard derivative-based triggering performs efficiently for isolated events but it may fail when secondary pulses lie on the decaying tail of a preceding pulse. To address this limitation, we introduce the Delayed Derivative Subtraction (DDS) method, which removes the local mean of the signal derivative prior to thresholding. We evaluate DDS using extensive simulations of pulse pairs generated with the official X-IFU simulator `xifusim` and reconstructed with the `SIRENA` software. Parameter scans across energies (0.2-12 keV) and separations (1-120 samples) show that DDS substantially enhances secondary-pulse detection efficiency without added computational complexity. An optimal configuration of a two-sample window ($w = 2$) and one-sample offset ($o = 1$) yields a global detection efficiency of 95%, compared with 70% for the standard algorithm. DDS provides an effective and lightweight solution for improving trigger robustness in the X-IFU readout chain and is now implemented in the publicly available SIRENA reconstruction package.

1. Introduction

The X-ray Integral Field Unit (X-IFU; Barret et al., 2023; Peille et al., 2025) will provide unprecedented high-resolution cryogenic imaging spectroscopy for Athena. Its core detectors are Transition Edge Sensor (TES) microcalorimeters, which measure the energy of individual X-ray photons through thermally induced current pulses in a voltage-biased circuit.

The on-board Event Processor identifies photon events using a standard triggering scheme in which a time derivative is applied to emphasize the rising edges of TES pulses (Boyce et al. 1999). A trigger is issued when this filtered derivative exceeds a threshold previously established from noise-only data streams.

Although effective for well-separated pulses, this method becomes unreliable in high count-rate scenarios. When two pulses arrive in close succession, the derivative

of the second pulse may be suppressed by the primary pulse's falling tail, causing the secondary event to be missed—especially for low-energy photons (Cobo et al. 2018).

To address this limitation, we develop and evaluate an enhanced triggering scheme: Delayed Derivative Subtraction (DDS). DDS aims to increase triggering robustness in the presence of overlapping pulses while maintaining minimal on-board computational complexity.

2. Baseline Pulse detection

The standard pulse detection process computes a time derivative (d_i) of the raw TES current signal (s_i). In this work we adopt a one-sided (causal) differentiator based on the Holoborodko smooth differentiator[1] ($N = 3$), characterized by the coefficient set $[1, 1, -1, -1]$ ($d_i = \sum_{k=0}^{3} C_k \cdot s_{i-k}$). This structure efficiently estimates the derivative while suppressing high-frequency noise near the Nyquist limit. The current baseline X-IFU implementation includes a one-sample shift to ensure that detection does not incur an additional sampling delay.

A pulse is triggered when a required number of contiguous derivative samples exceed a fixed threshold and the trigger resets only when the derivative falls again below that threshold. While simple and fast, this method struggles in pile-up conditions: if a secondary pulse arrives before the derivative of the primary has returned to baseline, it may fail to cross the threshold and thus remain undetected (Cobo et al. 2018).

3. Optimized Detection: Delayed Derivative Subtraction (DDS)

To improve triggering performance in complex pulse environments, we introduced the Delayed Derivative Subtraction method. DDS modifies the standard derivative by subtracting a local mean computed over a window preceding the candidate trigger point.

$$d_{i,\mathrm{dds}} = d_i - \frac{1}{w} \sum_{j=1}^{w} d_{i-j-o}, \tag{1}$$

where w is the subtraction window size and o is the offset between the trigger point and the averaging window. By compensating for the residual slope caused by the decaying tail of preceding pulses, DDS effectively restores the derivative baseline. As a result, small or moderate-energy secondary pulses remain detectable even in the presence of substantial overlap.

A comparison of the standard and DDS derivatives is presented in Figure 1.

4. DDS performance evaluation

We conducted a systematic performance study of DDS using synthetic pulse pairs generated with the official X-IFU simulator `xifusim` (Kirsch et al. 2022). Energies for the primary (E_1) and secondary (E_2) pulses were varied across the full X-IFU band

[1] http://www.holoborodko.com

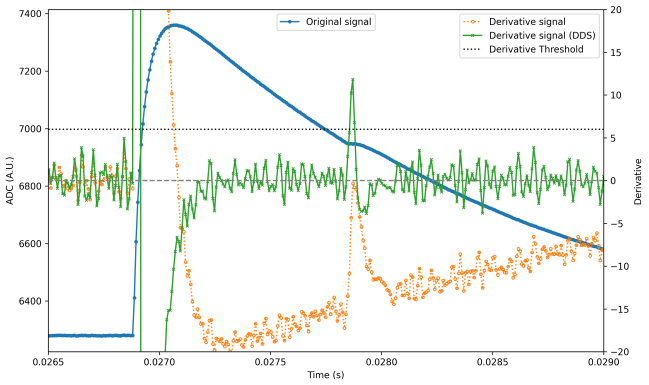


Figure 1. Simulated pair of closely spaced X-IFU pulses (solid line with filled circles) compared with their standard derivative (dotted line with open circles) and DDS derivative (solid line with '×' symbols). The DDS processing restores the derivative baseline, remaining above the trigger threshold at the arrival of the second pulse and enabling its detection.

(0.2–12 keV). Pulse separations ranged from 1 to 120 samples (7.7μs per sample), covering both extreme pile-up and well-separated conditions.

All datasets were then processed with `SIRENA`[2] (Ceballos et al. 2017) updated to incorporate DDS. For each parameter combination, the detection probability of the secondary pulse was computed, generating three-dimensional detection-efficiency cubes parameterized by E_1, E_2, and separation.

A full parameter scan was performed over window sizes $w = 1-6$ and offsets $o = 0-6$ samples. For each (w, o) pair, detection cubes were averaged first over all energy combinations and then over separations to derive a global performance metric.

The optimal configuration was determined to be a subtraction-window size of $w = 2$ samples and an offset of $o = 1$ sample. Under these settings, DDS achieved an average global detection efficiency of 95%, significantly outperforming the standard algorithm's 70%. This improvement is clearly illustrated in the detection-performance map in Figure 2, where the standard method corresponds to the configuration $w = 0$ samples and $o = 0$.

5. Conclusions

We present the DDS triggering method, a lightweight and effective enhancement to the standard derivative-based pulse-detection scheme currently baselined in the Athena X-IFU readout chain. By subtracting the local mean of the derivative signal over a delayed window, DDS compensates for slope biases caused by pulse tails, enabling the reliable detection of closely spaced events.

Extensive simulations across energies and pulse separations demonstrate that DDS significantly improves detection efficiency under high count-rate conditions, with an optimal configuration of subtracting-window size of $w = 2$ and offset of $o = 1$. Impor-

[2] `https://github.com/bcobo/SIRENA`

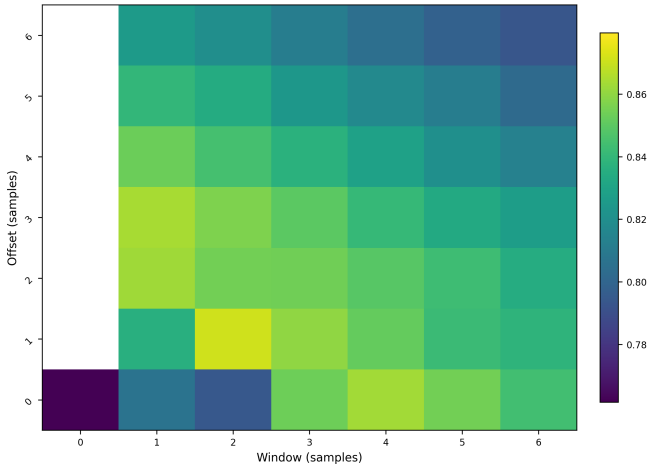


Figure 2. Detection efficiency of the DDS algorithm as a function of subtraction-window size w and offset o, derived from simulated X-IFU pulse pairs over the full energy and separation parameter space. The configuration $w = 0$, $o = 0$ corresponds to the standard derivative method and is shown for comparison.

tantly, DDS requires negligible additional computational effort, preserving compatibility with the resource constraints of the on-board Event Processor.

DDS is now implemented in the publicly available SIRENA reconstruction software and provides a robust solution for improving X-IFU event triggering.

Acknowledgments. The authors thank funding by grant PID2021-122955OB-C41 and grant PID2024-155779OB-C31 funded by MICIU/AEI/10.13039/501100011033 and by ERDF/EU.

References

Barret, D., et al. 2023, Experimental Astronomy, 55, 373. URL https://ui.adsabs.harvard.edu/abs/2023ExA....55..373B

Boyce, K. R., et al. 1999, in EUV, X-Ray, and Gamma-Ray Instrumentation for Astronomy X, edited by O. H. W. Siegmund, & K. A. Flanagan (SPIE), vol. 3765, 741 . Backup Publisher: International Society for Optics and Photonics, URL https://doi.org/10.1117/12.366557

Ceballos, M. T., et al. 2017, in ADASS XXV, edited by N. P. F. Lorente, K. Shortridge, & R. Wayth (San Francisco: ASP), vol. 512 of ASP Conf. Ser., 605. URL https://aspbooks.org/custom/publications/paper/512-0605.html

Cobo, B., et al. 2018, in Space Telescopes and Instrumentation 2018: Ultraviolet to Gamma Ray, edited by J.-W. A. den Herder, S. Nikzad, & K. Nakazawa, International Society for Optics and Photonics (SPIE), vol. 10699, 106994S. URL https://doi.org/10.1117/12.2313268

Kirsch, C., et al. 2022, Journal of Low Temperature Physics. URL https://link.springer.com/10.1007/s10909-022-02700-4

Peille, P., et al. 2025, Experimental Astronomy, 59, 18. URL https://link.springer.com/10.1007/s10686-025-09984-w

Astronomical Data Analysis Software and Systems XXXV
*ASP Conference Series, Vol. **VOL** Compiled: Tuesday 30th June, 2026@ 01:30*
K. Polsterer, D. Bohlender, S. Gwyn, A. Hincks and P. Teuben, eds.
*© **YEAR** Astronomical Society of the Pacific*

ZZZDEBUG!004.tex
Cardiel, N.|textbf
Chillarón-Víctor, M.
Pascual, S.
Gallego, J.
packages!Megaradrp
computer
languages!Python

A new algorithm for the detection and correction of cosmic rays in MEGARA exposures at the Gran Telescopio Canarias

Nicolás Cardiel,[1,2] María Chillarón-Víctor,[1,2] Sergio Pascual,[1,2] and Jesús Gallego[1,2]

[1]*Departamento de Física de la Tierra y Astrofísica, Facultad CC. Físicas, Universidad Complutense de Madrid, E-28040 Madrid, Spain;* cardiel@ucm.es

[2]*Instituto de Física de Partículas y del Cosmos, IPARCOS, Facultad CC. Físicas, Universidad Complutense de Madrid, E-28040 Madrid, Spain*

Abstract. We present the specific procedure designed to detect and correct cosmic ray hits in astronomical images obtained with the MEGARA integral field spectrograph at the Gran Telescopio Canarias. Traditional approaches in the MEGARA data reduction pipeline rely on median stacking of multiple exposures to mitigate CR contamination. However, this method becomes less effective for long exposures, where the same pixel may be struck by cosmic rays in consecutive frames. The key innovation of this work lies in the implementation of a strategy specifically tailored for the analysis of equivalent exposures heavily affected by cosmic rays. This approach employs numerical simulations derived directly from the observational data to estimate the detection probability of a cosmic ray as a function of signal level. The methodology allows the identification of cases in which the same pixel is affected in more than half of the exposures, thereby enabling a mean combination that yields an improved signal-to-noise ratio compared to the traditional median-based approach.

1. Introduction

One of the major challenges in processing CCD data is contamination by cosmic rays (CRs), high-energy particles, primarily protons and atomic nuclei, that originate from space and interact with the detector during exposure. When a CR strikes a CCD, it deposits a significant amount of energy in a localized region, producing spurious signals that do not correspond to real astronomical sources. The problem has been acknowledged since the earliest use of CCDs in astronomical imaging (e.g., Goad 1980).

Cosmic ray hits typically appear as sharp, bright pixels or elongated streaks, depending on the particle's trajectory and energy (Smith et al. 2002; Groom 2004). Unlike photon-induced signals, which follow the point-spread function of the telescope optics, CR artifacts are highly irregular and can mimic or obscure faint sources. This contamination becomes particularly problematic in long-exposure observations, where the probability of CR impacts increases with exposure time. For instruments operating outside Earth's atmosphere, such as space telescopes, the rate of CR hits is even higher due to the absence of atmospheric shielding (Murtagh & Adorf 1991; STScI 2008).

Over the last decades numerous strategies have been proposed to identify and correct cosmic-ray-affected pixels. The most straightforward approach to CR rejection

involves acquiring multiple exposures of the same field and identifying outlier pixels through statistical comparison (e.g., Windhorst et al. 1994). A median image calculated from aligned single exposures effectively creates a cosmic-ray-free reference, against which each exposure can be compared to identify contaminated pixels.

Removal strategies in individual exposures typically exploit the difference in morphology and intensity of the affected CR pixels in comparison with those actually exhibiting the signal of real astronomical sources. In this sense, a very popular method is the Laplacian edge-detection algorithm, commonly known as `L.A.Cosmic`, that was introduced by van Dokkum (2001). This method detects CRs as regions of high local curvature in the Laplacian-transformed image, efficiently distinguishing them from undersampled stellar profiles.

2. Removing cosmic rays in IFU exposures: the MEGARA case

When dealing with Integral Field Unit (IFU) exposures, traditional CR detection methods like `L.A.Cosmic` face several challenges that reduce their effectiveness. The raw IFU spectral images show fiber spectra distributed in a spatially scrambled pattern across the detector, where CR detection must occur before spectral extraction and cube assembly. Since algorithms like `L.A.Cosmic` were designed for direct imaging, they do not encounter the expected spatial coherence patterns in IFU data. Additionally, the tightly packed fiber spectra mean that a single CR hit can affect multiple adjacent fibers simultaneously, creating contamination patterns different from the sharp, isolated features these algorithms were designed to detect. Furthermore, legitimate astronomical features such as emission lines and absorption features produce sharp edges along the spectral direction that can mimic CR signatures, making discrimination difficult.

In this work, we focus on the identification and removal of CR events affecting CCD exposures obtained with MEGARA, the fiber-fed optical IFU installed at the Gran Telescopio Canarias (Gil de Paz et al. 2018; Carrasco et al. 2018). The usual operation of this instrument consists of taking three identical exposures of the same field in order to remove CRs using a median combination. An important problem we have encountered is that in typical long (~ 1800 s) exposures there are on the order of $\sim 10^2$ pixels affected by CRs in two of the three available exposures. These pixels compromise the scientific analysis of the reduced data.

3. The `M.M.Cosmic` algorithm

The proposed method builds upon fundamental principles that have been widely employed in previous studies addressing the expected signal characteristics of CCD detectors for a given gain and readout noise. Several works have applied these same fundamental principles to the identification of CRs in astronomical images, implementing different algorithms to optimize detection and removal.

The new algorithm, named M.M.Cosmic, implemented in the module `crmasks`[1] within the Python `numina` package (Pascual et al. 2025), introduces a strategy that identifies anomalous pixel values using a diagnostic diagram based on the median and

[1] `https://guaix-ucm.github.io/numina-tools/crmasks/crmasks.html`

minimum signal levels across three or more equivalent exposures. The method generates CR masks that can be applied using different strategies, each offering different trade-offs between noise suppression and CR removal efficiency. The `M.M.Cosmic` method consists of the following steps (assuming 3 equivalent exposures are available):
Step 1: Generation of a 3D stack with the three individual image arrays.
Step 2: Collapse the 3D stack to compute the `min2d`, `median2d`, `mean2d` and `max2d` bidimensional images.
Step 3: Generate a diagnostic diagram (see Fig. 1), comparing `(median2d-min2d)` vs. `(min2d-bias)`. In this diagram, pixels affected by CRs in more than half of the exposures exhibit a very high median signal relative to the minimum value reached in those pixels. A detection boundary is then computed using Monte Carlo simulations. To achieve this, the median image is employed to generate synthetic exposures, which are then used to build a 2D histogram representing the expected number of pixels in each small region of the diagnostic diagram (left panel in the figure below). The envelope that encloses all simulated points from above becomes the detection boundary. The same boundary is applied to the 2D histogram generated from the real images (right panel) to flag the suspicious pixels, whose signal is replaced by the value in min2d or in the corrected image provided by the `L.A.Cosmic` algorithm.

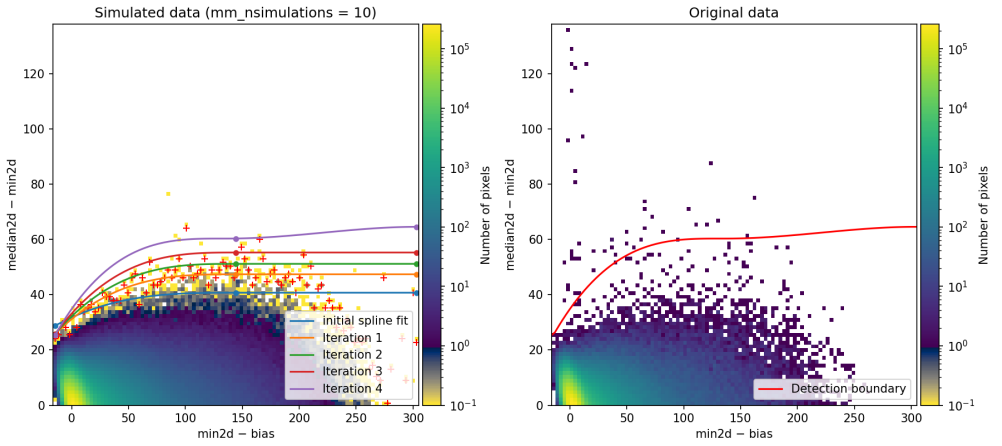


Figure 1. Simulated (*left*) and real (`right`) Minimum–Median diagnostic diagram.

Step 4: The same detection boundary can be used in a variant of the diagnostic diagram in which `(mean2d-min2d)` is plotted against `(min2d-bias)`. In this case, the goal is to identify pixels suspected of having been affected by CRs in the mean2d image, which contains all CRs from all individual exposures.
Step 5: We create diagnostic diagrams representing `(image#i-min2d)` against `(min2d-bias)`. This allows CR detection masks to be generated for each individual exposure.

The `max2d` array helps to reject false CR detections at low signal levels. We complement the detections made with `M.M.Cosmic` with those obtained using `L.A.Cosmic`: there are some CRs that are detected by only one of the two methods.

The different masks generated in steps 3 to 5 can be used to create a median image free of coincident CRs, as well as to attempt to produce a mean combination that removes all the CRs from the mean image. This latter option improves the signal-to-noise ratio, since the maximum information is used in pixels that are not affected by CRs.

The algorithm can be run independently of the MEGARA DRP by using the Python script `numina-crmasks`.

At the time of writing, we have just implemented the use of the PyCosmic method (Husemann et al. 2012) instead of L.A.Cosmic, and the first tests seem to indicate that it performs better in Steps 4 and 5 described above.

We plan to continue refining the described algorithm and intend to use it to remove CRs in the observations with the future instrument TARSIS (Gil de Paz et al. 2024), a wide-FoV IFU, proposed for the Calar Alto 3.5m telescope.

Acknowledgments. We thank the support of the projects PID2021-123417OB-I00 and PID2022-138621NB-I00, funded by MCIN/AEI/10.13039/501100011033/FEDER, EU, and PCI2022-135023-2, funded by MCIN/AEI/10.13039/501100011033 and by the European Union 'NextGenerationEU'/PRTR. This work made use of `Astropy` (Astropy Collaboration et al. 2013, 2018), `Numpy` (Harris et al. 2020), `Scipy` (Virtanen et al. 2020), `ccdproc` (Craig et al. 2017), and `Matplotlib` (Hunter 2007).

References

Astropy Collaboration, Price-Whelan, A. M., et al. 2018, AJ, 156, 123. eprint: 1801.02634

Astropy Collaboration, Robitaille, T. P., et al. 2013, A&A, 558, A33. eprint: 1307.6212

Carrasco, E., et al. 2018, in Ground-based and Airborne Instrumentation for Astronomy VII, edited by C. J. Evans, L. Simard, & H. Takami, vol. 10702 of Society of Photo-Optical Instrumentation Engineers (SPIE) Conference Series, 1070216

Craig, M., et al. 2017, astropy/ccdproc: v1.3.0.post1. URL `https://doi.org/10.5281/zenodo.1069648`

Gil de Paz, A., et al. 2018, in Ground-based and Airborne Instrumentation for Astronomy VII, edited by C. J. Evans, L. Simard, & H. Takami, vol. 10702 of Society of Photo-Optical Instrumentation Engineers (SPIE) Conference Series, 1070217

— 2024, in Ground-based and Airborne Instrumentation for Astronomy X, edited by J. J. Bryant, K. Motohara, & J. R. D. Vernet, vol. 13096 of Society of Photo-Optical Instrumentation Engineers (SPIE) Conference Series, 1309620

Goad, L. E. 1980, in Conference on Applications of Digital Image Processing to Astronomy, edited by D. A. Elliott, vol. 264 of Society of Photo-Optical Instrumentation Engineers (SPIE) Conference Series, 136

Groom, D. 2004, in Scientific Detectors for Astronomy, edited by P. Amico, J. W. Beletic, & J. E. Beletic (Dordrecht: Springer Netherlands), 81

Harris, C. R., et al. 2020, Nat, 585, 357. eprint: 2006.10256

Hunter, J. D. 2007, Computing in Science & Engineering, 9, 90

Husemann, B., Kamann, S., Sandin, C., Sánchez, S. F., García-Benito, R., & Mast, D. 2012, A&A, 545, A137. eprint: 1208.1696

Murtagh, F. D., & Adorf, H. M. 1991, in ESO Conf. and Workshop Proceedings, edited by P. J. Grosbøl, & R. H. Warmels, vol. 38 of ESO Conf. and Workshop Proceedings, 51

Pascual, S., Cardiel, N., & Picazo-Sánchez, P. 2025, guaix-ucm/numina: Release v0.36.0. URL `https://doi.org/10.5281/zenodo.17554628`

Smith, A. R., McDonald, R. J., Hurley, D. C., Holland, S. E., Groom, D. E., Brown, W. E., Gilmore, D. K., Stover, R. J., & Wei, M. 2002, in Sensors and Camera Systems for Scientific, Industrial, and Digital Photography Applications III, edited by M. M. Blouke, J. Canosa, & N. Sampat, vol. 4669 of SPIE Conference Series, 172

STScI 2008, WFPC2 Instrument Handbook (STScI), version 10.0 ed. URL `https://www.stsci.edu/hst/wfpc2/documents/handbook`

van Dokkum, P. G. 2001, PASP, 113, 1420. eprint: astro-ph/0108003

Virtanen, P., et al. 2020, Nature Medicine, 17, 261. eprint: 1907.10121

Windhorst, R. A., Franklin, B. E., & Neuschaefer, L. W. 1994, PASP, 106, 798

Astronomical Data Analysis Software and Systems XXXV
*ASP Conference Series, Vol. **VOL** Compiled: Tuesday 30th June, 2026@ 01:30*
K. Polsterer, D. Bohlender, S. Gwyn, A. Hincks and P. Teuben, eds.
*© **YEAR** Astronomical Society of the Pacific*

ZZZDEBUG!007.tex
Demleitner, M.|textbf
databases!query language!ADQL
software!packages!DaCHS
Virtual Observatory (VO)!standards
DaCHS! ascl:1804.005

Less Cumbersome Units in ADQL

Markus Demleitner[1]

[1] *Universität Heidelberg, Zentrum für Astronomie;*
msdemlei@ari.uni-heidelberg.de

Abstract. Since ADQL 2.1, there is a function IN_UNIT that converts a dimensional numeric expression to a user-provided unit as in, say, IN_UNIT(pmra, 'mas/yr'). While this already is a valuable tool for writing robust und portable queries, it is a lot to type, and the many letters and special characters make queries hard to read when multiple unit coercions are requested. Also, there is no way to have literals with units in this scheme.

In this contribution, I will propose a more compact syntax that allows one to write literals with units as in 3600{mas/deg} and to coerce expressions into query-specified units as in (rv/2)@{pc/yr}. I will also discuss the rationale for this design as well as alternatives that might be considered.

1. Introduction

Originally defined in the late 2000s, the first version of the Astronomical Data Query Language ADQL (Osuna et al. 2008) – which had a version tag of 2.0 despite being the first actual recommendation due to a rather bumpy standardisation process – did not contain any facilities for dealing with units. This clearly was unsatisfactory for a language designed to express physics; being explicit about units will make queries a lot more robust and portable between different tables and services, and it will also improve metadata quality, in particular when complex expressions end up in the select list.

Hence, early on there were initiatives to provide a feature to coerce columns or expressions to a certain unit. For instance, the TAP Implementation Notes (Demleitner et al. 2013) proposed a function-like construct:

> ADQL translators can typically introspect the tables they operate on, and thus can typically infer the (physical) unit of a column. Manually converting units (as in col_in_deg*3600) is error-prone, and expressions like that make it almost impossible to infer the unit of the result.
>
> This problem is addressed by the introduction of a function IN_UNIT(expr, <character_string_literal>); the second argument has to be a literal in order to make sure that an ADQL translator has access to its value; this value must be in the format defined by [a pre-release version of Derriere et al. (2014)]. The intended functionality is that the translator replaces the function call with an new expression that is *expr* given in the unit defined by the second argument if the translator can figure out *expr*'s unit, and it knows how to convert values in one unit into another. In every other case, the query must be rejected as erroneous.

This proposal, in essence, found its way into ADQL 2.1 (Mantelet et al. 2023), so that with modern ADQL engines you can write epoch propagation code like this:

```
ivo_epoch_prop(ra, dec, IN_UNIT(parallax, 'mas'),
IN_UNIT(pmra, 'mas/yr'), IN_UNIT(pmdec, 'mas/yr'),
IN_UNIT(rv, 'km/s'), 2024.4, 2000.0)
```

One will still have to adapt column names and the source epoch when applying such a query to a different catalogue, but the machine will take care that once these updates are made (for instance, by inspecting UCDs or using suitable data model annotation), the units of the function arguments will be as expected by the `ivo_epoch_prop` user defined function defined in Juaristi Campillo et al. (2024).

However, with the existing `IN_UNIT`, even relatively simple expressions become hard to read, and there are no provisions for units on literals at all. The use case motivating the original proposal from the implemenation note, `col_in_deg*3600`, in this sense still cannot be covered: there is no way the ADQL engine could infer that the result is in arcsec.

2. Less Cumbersome Unit Specifications

2.1. Two New ADQL Symbols

To provide more comprehensive and accessible unit facilities in ADQL, we suggest two new ADQL constructs:

- a non-terminal *unit-literal*, which is a string of characters not containing curly braces and itself delimited by curly braces. The material between the curly braces is required to be a VOUnits (Gray et al. 2023) string (with some further constraints discussed in sect. 2.2), but that requirement would be verbal and not part of the ADQL grammar.
- an operator *unit-coercion*, for which @ seems an appropriate literal.

A *unit-literal* would be allowed

- after an *unsigned-numeric-literal* without an operator (semantics: literal with a unit).
- after the *unit-coercion* operator; the left-hand side of *unit-coercion* would be a *factor* (semantics: unit conversion)[1]

I believe this will cover our use cases and makes for a fairly minimal change in ADQL's syntax. It lets an ADQL engine figure out with certainty that the result of

```
(mass@{solMass}+0.5{solMass})@{kg}
```

[1] This is a deviation from the poster version of this contribution, which suggested *numeric-value-expression* as the lhs. We reconsidered in order to make *unit-coercion* bind more tightly than the multiplicative operators.

is the original mass plus half a solar mass, given in kg regardless of what mass is actually in. It also would cover the original motivating use case via

```
col*3600{arcsec/deg}
```

Incidentally, this latter expression is different from the more straightforward related formulation `col_in_deg@{arcsec}` in that if col were in arcminutes, the resulting unit would be `arcmin.arcsec/deg`. This may appear rather useless in this particular context but may actually be helpful elsewhere.

2.2. Error Conditions

Unit annotation and conversion can fail for a number of reasons, including:

- The input expression is not in VOUnits syntax, or it contains what VOUnits calls an unknown unit (these are written in single quotes).
- The unit of the original expression is not known, either because of insufficient metadata (e.g., from an uploaded table), or because it cannot be inferred (e.g., when the expression is the sum of two quantities with different units, which as such is not defined as an error condition by ADQL and probably should not be, either).
- A conversion between incommensurable units is requested (e.g., Kilograms to Joules).

If a service cannot perform a conversion for any of these (or other) reasons, it is a fatal error; we do not want to introduce new sources of silent errors.

Only conversions between strictly commensurable quantities are supported. This in particular concerns conversions between spectral quantities given as energy, wavelength, frequency, or wave number. For the purposes of this proposal, all these are mutually incommensurable. This mechanism thus is not intended to be or grow into a replacement for the proposed `ivo_specconv` user defined function envisioned for version 1.3 of the calalogue of user defined functions (Juaristi Campillo et al. 2024).

While services should make a best effort to unit-annotate ADQL parse trees and figure out conversions, there is no strict requirement to do so because it is hard to delineate what minimal compliance would comprise. A trivial and valid implementation of the proposed mechanism would thus be to simply reject all conversions and to discard unit annotations. But of course, that would not yield a good user experience.

3. Alternative Approaches Considered

There are several other ways we considered for a more comprehensive and streamlined support of phyiscal units in ADQL:

- Simply writing units after literals would require an inclusion of the VOUnits grammar with all its terminals inside the ADQL grammar. Even if this were considered proportionate, we could not distinguish between a unit and a column name from an as-clause without the (optional) AS. In consequence, all terminals in VOUnits would need to become ADQL reserved words, which would be a massively breaking change.

- We could allow unadorned string literals after *unit-coercion* and *unsigned-numeric-literal*. This probably does not conflict with other legal ADQL constructs, but having distinct characters for unit literals helps producing useful error messages, and we save the need to escape single quotes; these are legal in VOUnits, albeit only for the unknown units that are not very useful in ADQL.

- We could define a fake schema (u, say) that contains the unit literals, so users would write 200*u.km/u.s. That syntax would have the advantage that many users are familiar with it from astropy. But it would require a custom VOUnits parser of its own, and it is unclear how we would deal with VOUnits functions in this scheme.

4. Implementation Status

This proposal can be tried on the GAVO's Heidelberg TAP service at `http://dc.g-vo.org/tap`. It will be part of release 2.13 of the DaCHS VO server package (Demleitner et al. 2014). The initial implementation (presumably still with problems) happened in commit 57539d54 of DaCHS' git repository[2].

Acknowledgments. This work was supported by BMBF FKZ 05A23VH4 ("e-inf-astro").

References

Demleitner, M., Harrison, P., & Taylor, M. 2013, TAP Implementation Notes Version 1.0, IVOA Note 13 December 2013

Demleitner, M., Neves, M. C., Rothmaier, F., & Wambsganss, J. 2014, Astronomy and Computing, 7, 27. eprint: 1408.5733

Derriere, S., Gray, N., Demleitner, M., Louys, M., & Ochsenbein, F. 2014, Units in the VO Version 1.0, IVOA Recommendation 23 May 2014. eprint: 1509.07267

Gray, N., Cecconi, B., Demleitner, M., Derrière, S., Gray, N., Louys, M., & Ochsenbein, F. 2023, Units in the VO Version 1.1, IVOA Recommendation 15 December 2023

Juaristi Campillo, J., Demleitner, M., & Civera, T. 2024, Catalogue of ADQL User Defined Functions Version 2.2, IVOA Endorsed Note 07 November 2024

Mantelet, G., Morris, D., Demleitner, M., Dowler, P., Lusted, J., Nieto-Santisteban, M. A., Ohishi, M., O'Mullane, W., Ortiz, I., Osuna, P., Shirasaki, Y., & Szalay, A. 2023, Astronomical Data Query Language Version 2.1, IVOA Recommendation 15 December 2023

Osuna, P., Ortiz, I., Lusted, J., Dowler, P., Szalay, A., Shirasaki, Y., Nieto-Santisteban, M. A., Ohishi, M., O'Mullane, W., VOQL-TEG Group, & VOQL Working Group. 2008, IVOA Astronomical Data Query Language Version 2.00, IVOA Recommendation 30 October 2008. eprint: 1110.0503

[2] `https://gitlab-p4n.aip.de/gavo/dachs`

Astronomical Data Analysis Software and Systems XXXV
*ASP Conference Series, Vol. **VOL** Compiled: Tuesday 30th June, 2026@ 01:30*
K. Polsterer, D. Bohlender, S. Gwyn, A. Hincks and P. Teuben, eds.
*© **YEAR** Astronomical Society of the Pacific*

ZZZDEBUG!009.tex
Zhang, Y.|textbf
Zhang, J.
Li, C.

Targeting YSOs from Large Sky Surveys by Machine Learning

Yanxia Zhang,[1] Jingyi Zhang,[1] and Changhua Li[1]

[1]*National Astronomical Observatories, Chinese Academy of Sciences, Beijing, China;* zyx@bao.ac.cn

Abstract. We review the application of machine learning (ML) to identify Young Stellar Objects (YSOs) in the era of large-scale astronomical surveys. It begins by outlining the limitations of traditional identification methods (e.g., infrared excess, spectroscopic confirmation), which struggle with the volume, complexity, and high-dimensionality of data from projects like the Vera C. Rubin Observatory (LSST) and JWST. Key challenges highlighted include severe class imbalance, as YSOs are rare, and contamination from other astrophysical sources. We present several case studies demonstrating the success of ML techniques. These examples show how algorithms like Gradient Boosting Machines, CatBoost combined with Self-Paced Ensemble (SPE), and Probabilistic Random Forests, especially when enhanced with methods like SMOTE to address class imbalance, have successfully identified large, pure samples of YSO candidates from multi-wavelength data. Looking forward, we predict that next-generation surveys will drive the adoption of more sophisticated models like Graph Neural Networks and Transformers. Future trends will include a shift towards physically informed ML that estimates parameters like mass and age, and a greater focus on model explainability using tools like SHAP to build trust and gain scientific insights. In conclusion, ML is revolutionizing YSO identification, turning the challenge of big data into an unprecedented opportunity for statistical studies of star formation.

1. Introduction

A Young Stellar Object (YSO) refers to a star in its early stages of evolution, occurring after its formation from the gravitational collapse of a molecular cloud but before it reaches the main sequence. This category encompasses both protostars and pre-main-sequence stars. Key characteristics include the presence of circumstellar accretion disks, highly collimated jets, and outflows, which are observable over timescales of a few years. YSOs are typically classified into evolutionary stages (Class 0, I, II, III) based on the slope of their Spectral Energy Distribution (SED), which reflects the amount and distribution of surrounding material such as envelopes and disks.

Before the widespread adoption of machine learning (ML), several methods were used to identify YSOs, each with notable strengths and weaknesses, as shown in Table 1.

The primary challenge these methods encounter in the era of large surveys such as the Vera C. Rubin Observatory (LSST) and the James Webb Space Telescope (JWST) is the trade-off between efficiency and scalability. While spectroscopic confirmation is highly reliable, it is not feasible for millions of targets. On the other hand, traditional

Table 1. Advantages and Challenges of Traditional Methods.

Method	Description	Advantages	Disadvantages / Challenges
Infrared Excess	Identifying YSOs by detecting excess infrared emission from circumstellar dust in disks and envelopes.	Directly probes the defining feature of YSOs. Effective for statistical studies.	Cannot easily distinguish YSOs from galaxies with similar IR signatures. Contamination from asymptotic giant branch stars is common.
Spectroscopic Features	Using optical/infrared spectra to identify features of youth, such as accretion-related emission lines (e.g., Hα) and lithium absorption.	Considered the "gold standard" for confirmation. Provides physical parameters (e.g., mass, age).	Extremely time-consuming for large surveys. Not feasible for the billions of sources in modern all-sky surveys.
Astrometric Selection	Using precise proper motions and parallaxes (e.g., from Gaia) to select stars co-moving within a known star-forming region.	Very effective for isolating members of a specific cluster. Reduces contamination from field stars.	Only works for stars in kinematically coherent groups. Cannot identify isolated YSOs in the field.

color-cut methods, although faster, lack the sophistication needed to effectively manage high-dimensional data and are prone to high contamination rates.

2. Key Challenges in Large-Scale YSO Identification

The shift to large sky surveys has amplified several key challenges:

- **Data Volume and Complexity**: Surveys such as ZTF and those conducted by Spitzer generate millions of light curves and SEDs, rendering manual inspection impractical.
- **Class Imbalance**: YSOs are exceedingly rare compared to other astrophysical sources, such as main-sequence stars and galaxies. Consequently, classifiers trained on imbalanced datasets tend to be biased toward the majority classes, leading to poor performance in accurately identifying these rare YSOs.
- **Dimensionality**: YSOs can be characterized by a multitude of features, including multi-band magnitudes, colors, variability indices, and astrometric measure-

ments. Traditional methods often struggle to effectively model these high-dimensional parameter spaces.

- **Contamination and Confusion**: Accurately distinguishing YSOs from other variable or infrared-bright sources – such as active galactic nuclei (AGN), Be stars, or field stars – remains a significant challenge when relying on limited photometric data.

3. Case Study: Machine Learning for YSOs

By comparing eight supervised ML algorithms, including tree-based models and neural networks, on multi-wavelength flux densities with 10-fold cross-validation, Miettinen (2018) addressed the challenge of rapidly classifying YSOs in Orion, achieving the highest accuracy (82%) with a Gradient Boosting Machine (GBM) to efficiently process large samples where traditional SED analysis is slow.

Based on tailored statistical learning methods applied to Spitzer/IRAC multi-band data, Kuhn et al. (2021) successfully identified approximately 120,000 YSO candidates across the Galactic midplane, effectively distinguishing YSOs from other red infrared sources despite challenges like source confusion and the similar SEDs of galaxies and YSOs at mid-infrared wavelengths.

By leveraging the CatBoost and Self-Paced Ensemble (SPE) ML methods on multi-wavelength data (X-ray, optical, infrared), Ma et al. (2024) effectively addressed the challenge of identifying rare YSOs from a large, combined survey catalog (XMM-WISE-SDSS), successfully discovering 412 new YSO candidates.

By applying the Synthetic Minority Oversampling Technique (SMOTE) to address class imbalance and the Probabilistic Random Forest (PRF) classifier to multiwavelength photometric data, Ghaziasgar et al. (2024) successfully categorized dusty stellar sources — including YSOs, CAGB, OAGB, RSG, and PAGB stars — in the Magellanic Clouds with high accuracy, demonstrating an effective ML framework for stellar population analysis in resolved galaxies.

By employing multi-wavelength near- and mid-infrared photometric data analyzed through specific color-color diagrams, the IdentYS tool addresses the challenge of efficiently identifying and classifying Class I and II YSOs in diverse star-forming regions while enhancing sample purity by filtering out field contaminants (Nikoghosyan et al. 2025).

4. Summary and Future Outlook

The identification of YSOs is fundamental to understanding star and planet formation. The field is undergoing a paradigm shift from traditional, often manual methods to data-driven, automated ML approaches to cope with the data deluge from modern sky surveys.

- **Next-Generation Surveys and Algorithms**: The full potential of ML will be realized with data from upcoming missions such as LSST and *JWST*. These advancements are likely to drive the adoption of more sophisticated models, such as Graph Neural Networks for modeling spatial relationships within star-forming

regions, and Transformer-based models for analyzing long-term temporal variability.

- **Physical Integration**: A significant trend will be the transition from pure classification to physically informed ML. This approach involves utilizing algorithms that not only classify a source as a YSO but also estimate physical parameters such as mass, age, and accretion rate. This could potentially be achieved by directly incorporating radiative transfer models into the learning process.
- **Focus on Explainability**: As models grow increasingly complex, it will be essential to enhance the interpretability of their decisions – utilizing tools such as SHAP (SHapley Additive exPlanations). This focus on explainability is crucial for gaining the trust of astronomers and for uncovering new scientific insights into the features that effectively differentiate between various classes of YSOs.

In conclusion, ML, especially techniques designed to handle imbalanced datasets like SPE combined with powerful classifiers like CatBoost, has revolutionized our ability to discover YSOs. This enables the construction of large, pure samples necessary for statistical studies of star formation across the galaxy, turning the challenge of big data in astronomy into an unprecedented opportunity.

Acknowledgments. This work is supported by the National Natural Science Foundation of China (NSFC; grant Nos.12273076, 12373110, 12133001), and the China Manned Space Program with grant no. CMS-CSST-2025-A06.

References

Ghaziasgar, S., Abdollahi, M., Javadi, A., van Loo, J. T., McDonald, I., Oliveira, J., & Khosroshahi, H. G. 2024, Communications of the Byurakan Astrophysical Observatory, 71, 377. eprint: 2504.14242

Kuhn, M. A., de Souza, R. S., Krone-Martins, A., Castro-Ginard, A., Ishida, E. E. O., Povich, M. S., Hillenbrand, L. A., & COIN Collaboration 2021, ApJS, 254, 33. eprint: 2011.12961

Ma, X., Zhang, Y., Zhang, J., Li, C., Kang, Z., & Li, J. 2024, AJ, 168, 210. eprint: 2410.11436

Miettinen, O. 2018, Ap&SS, 363, 197. eprint: 1808.08371

Nikoghosyan, E., Baghdasaryan, D., Andreasyan, D., Azatyan, N., Samsonyan, A., & Yeghikyan, A. 2025, Communications of the Byurakan Astrophysical Observatory, 72, 300. eprint: 2511.12696

Astronomical Data Analysis Software and Systems XXXV
*ASP Conference Series, Vol. **VOL** Compiled: Tuesday 30th June, 2026@ 01:30*
K. Polsterer, D. Bohlender, S. Gwyn, A. Hincks and P. Teuben, eds.
*© **YEAR** Astronomical Society of the Pacific*

ZZZDEBUG!010.tex
Gaillard, N.|textbf
Hernandez, J. L.
FOOBAR!conference!ADASS 2025
FOOBAR!organisations!ASP
FOOBAR! ascl:1101.010

Anomaly Detection for Astrometry in the Gaia Mission

Nelly Gaillard[1] and Jose Luis Hernandez [1]

[1]*European Space Agency - ESAC*

Abstract. This work introduces a framework consisting of statistical and machine learning models for the automated analysis of residuals, one of the output of Gaia calibration software. The goal is to detect anomalies that can be either indicators of non-convergence of the software, or problems in the calibration model.

1. Introduction

Gaia is a European space mission providing astrometry, photometry, and spectroscopy of nearly 2.7 billions stars in the Milky Way as well as significant samples of extragalactic and solar system objects. A fundamental component of the scientific data processing for the Gaia mission is the astrometric core solution, which determines the astrometric parameters (position, parallax, and proper motion) for the primary sources. At the same time, it provides the reference frame, represented by the instrument's attitude over time, and the geometric calibration of the instrument (Gaia collaboration 2016). This solution was implemented in a scientific software system named AGIS (Astrometric Global Iterative Solution), which solves a massive least-squares problem of around 500 million parameters (Lindegren et al. 2012). The solution is *global*, as a single least-square adjustment involves the complete set of parameters needed to describe the primary sources (a subset of the objects), the telescope attitude, and various instrument calibration parameters. The resulting system of equations is so large that its solution is only feasible with an *iterative* method.

One of the most challenging tasks that needs to be tackled in AGIS is creating a calibration model accurate enough to take into account the subtle effects that may impact the solution quality at the micro-arcsecond level. These effects may have a real physical origin, such as imperfections in the hardware or CCD manufacturing, or can be caused by limitations of the Point Spread Function models used to extract the basic input data for AGIS from Gaia's raw observations.

Among the AGIS outputs used to assess the quality of the solution there are residuals. These are the differences between the values predicted by the observation models and the actual observation data itself. Residuals are a collection of many data points (one for each Gaia observation) and depend on hundreds of parameters. They can be very noisy up to a certain level, but hidden in the noise one can find dependencies with some of the other accessible parameters (field of view, CCD, time, color, etc).

This paper introduces a framework for the automated analysis of residuals, to help the AGIS team uncover unknown dependencies or issues in the calibration model.

2. Anomalies in the Residuals

The residuals, as an output of AGIS, are expected to follow a normal distribution with zero mean and a standard deviation consistent with the photon noise for the corresponding magnitude. For a simpler evaluation, and a constant standard deviation over the magnitude, the target of the analysis are the *normalized residuals*, obtained by dividing the residuals by the observation sigma. However, in both cases, the real data exhibit clusters of observations with high absolute values and and recurring patterns for specific parameters. Some of these anomalous behaviours can be traced to known instrument issues, whereas others represent new discoveries.

The analysis of such anomalies can provide several benefits. First, it can uncover deficiencies in the calibration model, potentially leading to investigations into the causes of these biases, and subsequent corrections to better match the observations. Additionally, although many of the correlations of the residual parameters are known, a number of others remain hidden and become apparent only when the data are partitioned. When necessary, these dependencies can be modelled and incorporated into the calibration. Finally, this diagnostic process can reveal convergence issues of the AGIS solution. In this sense, the framework can serve as a quality assessment tool and assist in modifying the weighting scheme to improve algorithm convergence.

The nature of the AGIS post-fit residuals makes the Anomaly Detection (AD) task particularly challenging. In this context, the anomalies of interest are those that affect a consistent group of observations, the so called *group anomalies*. This means that individual outliers, while deviating from the nominal distribution, should not influence the analysis. A second challenge is the absence of ground truth, as most of the anomalous behaviours are not known a priori, making it impossible to use supervised approaches or to design approaches targeting specific types of anomalies. Moreover, the anomalies identified during the initial visual screening are highly heterogeneous, differing in shape, size, number of affected samples, average residual value.

These challenges make it difficult to develop a single general approach capable of detecting all types of anomalies simultaneously. Therefore, the proposed solution involves a suite of complementary methods, each addressing a different subset of anomalies.

3. Methods

The framework, implemented as a Python package, consists of three main approaches described below. Each addresses the anomaly detection task from a distinct perspective providing complementary results. The first method is based on a statistical thresholding scheme; the second leverages machine learning techniques to provide a qualitative characterization of anomalies; and the third focuses on identifying long-term systematic trends.

3.1. Statistical Method

The statistical method provides a simple yet robust and interpretable approach to anomaly detection, which serves as a rapid diagnostic tool. The underlying concept is to identify regions where outlier residuals are concentrated and to compare their statistical distribution with that of the nominal (inlier) sample. Areas where discrepancies between the two distributions exceed a predefined threshold are flagged as potentially anomalous.

The procedure is applied independently to each dimension of interest and develops as follows:

- The first step is to separate the observations based on the absolute value of normalized residuals, using a threshold of 2. Two sets of outliers are thus defined, when the normalised residuals are greater or smaller than 2 accordingly.
- Subsequently, to estimate the distribution over the dimension considered, for both subsets.
- Finally, to identify the regions in the considered dimension where the distributions of the nominal and anomalous data differ significantly.

The results can be visualized across the focal plane grid such as in figure 1 to highlight anomalous behaviors within or between CCDs. Thanks to its computational efficiency and ease of parallelization, this method provides a convenient diagnostic and supports targeted analyses of specific data partitions.

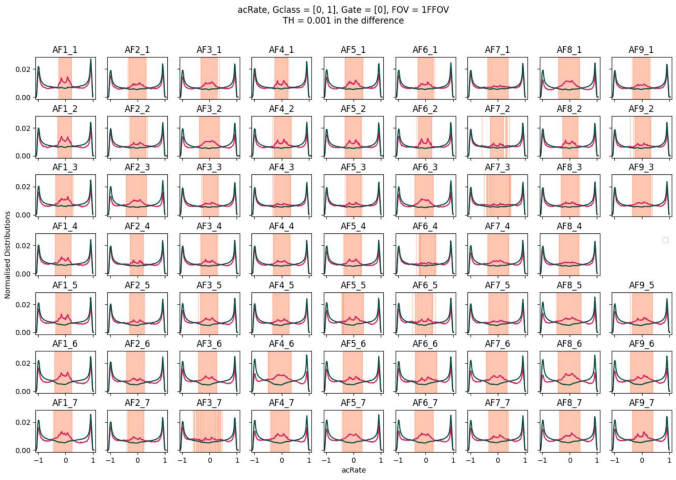


Figure 1. Focal plane grid, showing residuals distribution over acRate. Anomalous behaviours are highlighted

3.2. Machine Learning Method

While many anomalies are visually recognizable when inspecting data subsets along individual parameters, they become increasingly difficult to identify within the full multidimensional space. In this approach, the data are thus interpreted as time series along each relevant dimension for an efficient localization of group anomalies. This method comprises three consecutive stages:

- *Detection*: segments are generated from the time series via a moving window technique and processed with an Isolation Forest algorithm to identify anomalous windows. (Liu 2008). The size of the window is chosen experimentally for each dimension.

- *Clustering*: representative features are extracted from the anomalous segments (such as mean, max, min window values, number of peaks, etc.) and summarized to enable grouping with an HDBSCAN clustering algorithm.
- *Explanation*: to interpret the nature of each cluster, a classifier is trained to associate the original features with the cluster labels. The SHAP framework (Lundberg & Lee 2017), based on Shapley values and cooperative game theory, is then applied to determine which features influence the classification the most for each category.

This method offers qualitative insights into the properties and potential origins of anomalies. However, SHAP scores indicate the relative importance of features for the classifier's decisions rather than establishing causal relationships between variables and anomalies.

3.3. Long Trend analysis

This diagnostic step aims to assess whether the mean of normalized residuals deviates systematically from the nominal zero baseline along a given dimension. A simple linear fit is applied to the residual distribution; nominal data have slope coefficients close to zero, whereas cases exhibiting long term anomalies display slopes of higher absolute value. Due to its negligible computational cost, this method serves as an efficient screening tool to detect systematic deviations over a larger scale with respect to the previous methods. Although relatively coarse, its outcomes are straightforward to interpret and valuable for guiding further investigation.

4. Conclusions

The Framework here presented is a first step towards a comprehensive and efficient analysis of the AGIS post-fit residuals. Given the absence of ground truth and the unknown characteristics of each anomaly, its performance has been assessed by comparing the results of the three methods and through a manual inspection of representative anomaly samples from each detected class. The comparisons demonstrate strong consistency between methods, confirming the mutual reliability of the different approaches. Results are promising, as most of the anomalies detected were confirmed by AGIS experts. The Anomaly Detection Framework will be regularly used during the processing of Gaia Data Release 5 to assess and improve the quality of the astrometric solutions.

References

Gaia collaboration, T. e. a., Prusti 2016, Astronomy and Astrophysics, 595, A1. URL http://dx.doi.org/10.1051/0004-6361/201629272

Lindegren, L., Lammers, U., Hobbs, D., O'Mullane, W., Bastian, U., & Hernández, J. 2012, Astronomy and Astrophysics, 538, A78. URL http://dx.doi.org/10.1051/0004-6361/201117905

Liu, F. T. e. a. 2008, in Proceedings of the 2008 Eighth IEEE International Conference on Data Mining (USA: IEEE Computer Society), ICDM '08, 413–422. URL https://doi.org/10.1109/ICDM.2008.17

Lundberg, S. M., & Lee, S.-I. 2017, Advances in neural information processing systems, 30

Astronomical Data Analysis Software and Systems XXXV
*ASP Conference Series, Vol. **VOL** Compiled: Tuesday 30th June, 2026@ 01:30*
K. Polsterer, D. Bohlender, S. Gwyn, A. Hincks and P. Teuben, eds.
*© **YEAR** Astronomical Society of the Pacific*

ZZZDEBUG!019.tex
Bhatawdekar, R.|textbf
Arviset, C.
Merin, B.
Marchi, G.
Bruijne, J.
Baines, D.
Masson, A.
Bentley, M.
Perez, H.
Leon, I.
FOOBAR!conference!ADASS 2019
FOOBAR!organisations!ASP
FOOBAR! ascl:1101.010

More than Bytes: Technical and Social Dimensions of Data Lifecycle Management at ESA's Space Science Data Archives

Rachana Bhatawdekar,[1] Christophe Arviset,[1] Bruno Merin,[1] Guido de Marchi,[2] Jos de Bruijne,[2] Deborah Baines,[1] Arnaud Masson,[1] Mark Bentley,[1] Hector Perez,[3]and Ignacio Leon[3]

[1]*European Space Agency, ESAC, Villanueva de la Cañada, Madrid, 28691, Spain;* Rachana.Bhatawdekar@esa.int

[2]*European Space Agency, ESTEC, Keplerlaan 1, Noordwijk, 2201 AZ, The Netherlands*

[2]*Starion Group for ESA, Villanueva de la Cañada, Madrid, 28691, Spain*

Abstract. The long-term scientific value of space mission data depends not only on storage capacity and technical infrastructure, but also on governance, policies, standards, collaboration, and community engagement. At the European Space Agency's ESAC Science Data Centre (ESDC), data lifecycle management is approached as a holistic process spanning acquisition, processing, curation, access, reuse, and long-term preservation across more than thirty space science missions. This paper presents how ESDC combines technical and social dimensions to build sustainable, interoperable, and future-proof science archives. We describe the evolution of ESA's archive architecture, strategies for scaling to petabyte-scale multi-mission datasets, the role of interoperability standards, and recent initiatives such as the ESDC Multi-Mission Data Services (EMDS) platform and multi-mission archives. We also discuss how governance, cross-agency collaboration, and expertise retention are essential to sustaining archives beyond individual mission lifetimes. Finally, we reflect on lessons learned from recent technology shifts, illustrating how architectural modularity enables resilience in the face of unplanned change.

1. Introduction

Space science archives play a central role in maximising the long-term scientific return of space missions. While mission operations typically span years, archived data are expected to remain usable for decades, often long after the original technologies, software environments, and mission teams have disappeared. This creates unique challenges for archive infrastructures, which must balance innovation with stability, scalability with maintainability, and openness with governance.

In recent years, the astronomical community has increasingly recognised that data stewardship extends beyond data storage and access. Discoverability, interoperability, provenance, and reuse are now considered fundamental requirements for sustainable archives. The FAIR data principles, which stands for Findable, Accessible, Interoperable, and Reusable, provide a widely adopted framework for addressing these challenges

(Wilkinson et al. 2016). However, implementing FAIR principles in operational environments requires sustained technical investment and strong institutional coordination.

At the European Space Agency (ESA), the ESAC Science Data Centre (ESDC) operates a diverse portfolio of space science archives supporting a wide range of scientific disciplines and mission lifecycles. Rather than treating archiving as a purely technical activity, ESDC approaches data lifecycle management as an integrated socio-technical system, combining infrastructure, governance frameworks, international standards, and long term collaboration models. This paper presents ESDC's approach, focusing on how the interaction between technical and social dimensions enables archive sustainability and adaptability over decades.

2. The ESDC Archive Landscape

The ESAC Science Data Centre (ESDC) serves as the central hub for ESA's space science data archives and a major resource for the global scientific community. ESDC has hosted ESA archives since the late 1990s, beginning with the Infrared Space Observatory (ISO) Data Archive in 1998, and has since expanded significantly in both scale and scope.

As of 2026, ESDC hosts archives for more than thirty missions across four scientific domains: astronomy, planetary science, heliophysics, and human and robotic exploration. These include both archives actively developed for mission in operations, where software and services continue to evolve, and legacy mission archives that require long-term maintenance despite the end of mission operations. In 2024, the archives served approximately 35,000 monthly users, with over 100 TB of data downloaded per month and a total archive volume exceeding one petabyte.

The role of an archive at ESDC extends well beyond long-term data storage. Early archives, such as those for Ulysses and EXOSAT, were largely static repositories focused on data preservation and basic file distribution. In contrast, modern ESA archives are dynamic systems integrated with mission data processing pipelines and international scientific infrastructures. They support continuous ingestion, reprocessing, validation, and access through multiple user-facing and programmatic interfaces.

Today, ESDC archives function as full-fledged data management systems, providing discovery services, rich metadata, visualization capabilities, and interoperable access via mission-specific portals, cross-mission tools, application programming interfaces, and Virtual Observatory protocols. To manage growing data volumes and mission diversity, ESDC has progressively evolved toward a shared, multi-mission archive landscape, reusing common infrastructure and services wherever possible while retaining flexibility for mission-specific requirements. This balance between standardization and adaptability underpins the long-term sustainability of ESA's space science archives.

3. Data Lifecycle Management in Space Science

In space science archives, data lifecycle management is the structured handling of data from acquisition to long-term preservation and reuse. Unlike many other scientific domains, space mission data often remain valuable for decades, extending well beyond spacecraft operations, original processing systems, and mission teams.

The lifecycle begins with data acquisition by spacecraft instruments and processing at mission centres, where telemetry is transformed into calibrated, science-ready products (See Figure 1). During active mission phases, archives must support continuous ingestion, reprocessing, versioning, and provenance tracking as calibration pipelines evolve.

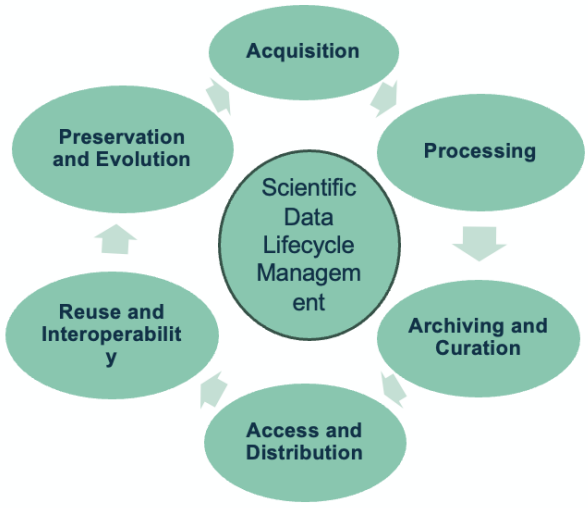


Figure 1. Data lifecycle management in space science.

Once archived, data are curated to ensure long-term usability and discoverability. At ESDC, access and distribution are provided through web interfaces, programmatic APIs, and international standards such as those defined by the Virtual Observatory, enabling efficient data discovery and retrieval.

A core objective of lifecycle management is reuse and interoperability. Scientific analyses increasingly rely on combining data from multiple missions and disciplines, enabled through standardized protocols such as Table Access Protocol (TAP) and Heliophysics API (HAPI), which provide consistent access across heterogeneous archives.

Finally, long-term preservation requires archives to evolve alongside changing technologies, data formats, and community practices. At ESDC, lifecycle considerations are integrated early in mission planning, ensuring sustainability, controlled evolution of archive systems, and maximized long-term scientific return.

4. Two Inseparable Dimensions: Technical and Social

At ESDC, data lifecycle management is explicitly framed around two inseparable dimensions.

The technical dimension encompasses infrastructure, architecture, scalability, interoperability, and long-term sustainability. It addresses how systems are designed, how data volumes are handled, and how services evolve over time.

The social dimension focuses on governance, policies, collaboration, and people. Technical systems alone cannot succeed without coordinated adoption of standards, shared operational practices, and sustained institutional knowledge. The intersection

of these dimensions is what enables ESDC to operate sustainable, trusted, and internationally interoperable data ecosystems.

5. Technical Dimensions of Data Lifecycle Management

5.1. Data Growth and Scalability

Over the past two decades, ESDC archives have grown from a few terabytes to approximately one petabyte, with projections exceeding 15 PB by the end of the decade (See Figure 2). This growth is accompanied by increasing mission diversity, each bringing unique data models, formats, and access patterns.

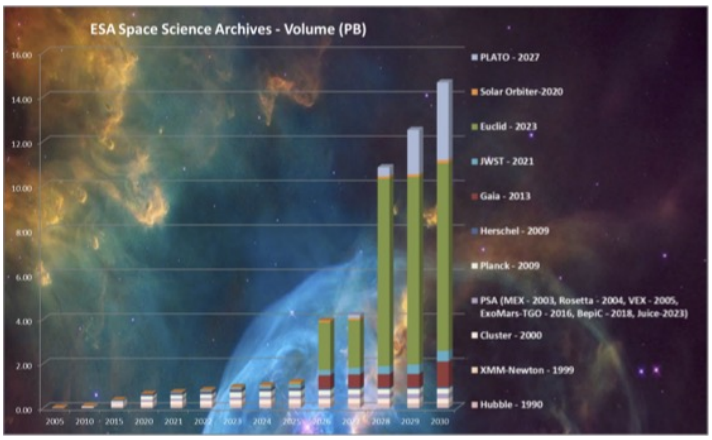


Figure 2. The growth of ESDC archives from a few terabytes to approximately one petabyte, with projections exceeding 15 PB by the end of the decade.

Scalability at ESDC therefore extends beyond storage capacity to include design, metadata management, and user interaction models. Traditional download-based workflows do not scale to petabyte-sized datasets, motivating a shift toward in-situ data analysis. We are supporting this shift through two complementary approaches:

- ESA Datalabs, a multi-mission science platform providing interactive environments (e.g. Jupyter notebooks) that allow users to analyse data close to where they are stored
- The Euclid Data Space (EDS) Portal, which integrates the Euclid Archive, ESA Datalabs, and ESASky into a unified environment for exploration, retrieval, and analysis.

Together, these initiatives reduce data movement, improve efficiency, and enable scalable scientific workflows.

5.2. Integrating Diverse Missions within a Unified Architecture

ESDC's archive architecture has evolved over more than two decades, guided by consistent principles of openness and modularity. Early mission archives, such as the ISO

archive in the early 2000s, already adopted a three-tier design separating data, services, and presentation layers, and introduced early Virtual Observatory (VO) concepts (Arviset et al. 2026) to enable external interoperability (See Figure 3).

While the underlying technologies have changed significantly, the architectural philosophy has remained stable. Today's mission archives, including the Euclid archive, are implemented as distributed systems. They are built technologies such as Greenplum, network attached storage, and VO compliant protocols including TAP, DataLink, and Simple Image Access (SIA). Over time, VO functionality has evolved from an external add on into a core architectural component (See Figure 3)

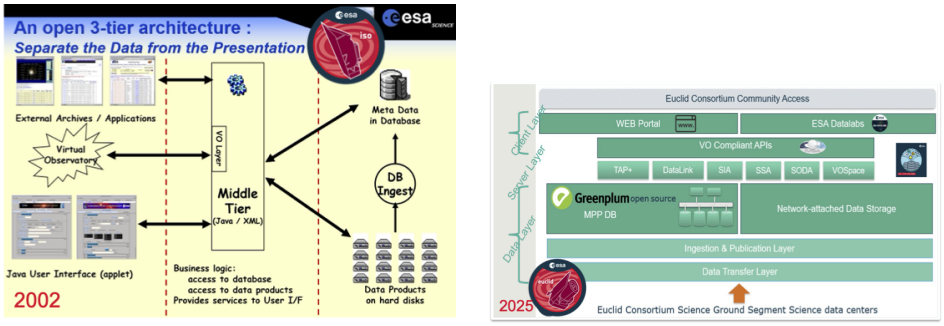


Figure 3. A stable architecture that evolved over time. *Left:* ISO archive in the early 2000s. *Right:* The architecture of Euclid archive today.

This architecture now supports more than thirty missions spanning astronomy, planetary science, heliophysics, and human and robotic exploration. All archives follow a consistent three-tier structure. The client layer provides mission-specific portals and cross-mission tools such as ESASky, as well as programmatic access via APIs and libraries such as `astroquery` and `SunPy`. The server layer exposes standardized protocols (e.g. TAP, ADQL, HAPI), enabling interoperable access across missions and disciplines. Beneath this, the data layer comprises mission databases, file storage, persistent identifiers, and robust ingestion pipelines.

Together, these elements form a mature, unified archive architecture that scales across scientific domains while supporting interoperability, long-term maintainability, and multi-mission data exploitation.

5.2.1. The EMDS Platform

A key recent evolution in this architecture is the ESDC Multi-Mission Data Services (EMDS) platform, introduced in 2024 (See Figure 3). EMDS provides shared infrastructure—storage, computing, and standardised interfaces—that allows multiple missions to be hosted on a common platform and represents a strategic shift toward shared, long-term infrastructure.

Architecturally, EMDS builds on the established three-tier model, exposing common data services that can be accessed through different user interfaces, ranging from mission-specific portals to sub-area and cross-mission applications. This design allows individual missions to evolve without introducing dependencies that would complicate long-term maintenance. It also supports different types of user interfaces, from mission-specific graphical user interfaces (GUIs) to sub-area portals, all connected to the same underlying data services.

By onboarding missions during their active phases, EMDS reduces the cost and complexity of later transitions to legacy operations. It also facilitates cross-mission services and consistent user support, aligning operational efficiency with scientific usability.

At the moment, the platform hosts products from SMILE and Einstein Probe mission. Looking ahead, additional archives will join the platform in production during 2026, including European X-ray Observatory SATellite (EXOSAT), ISO, Proba3, and the Heliophysics legacy missions.

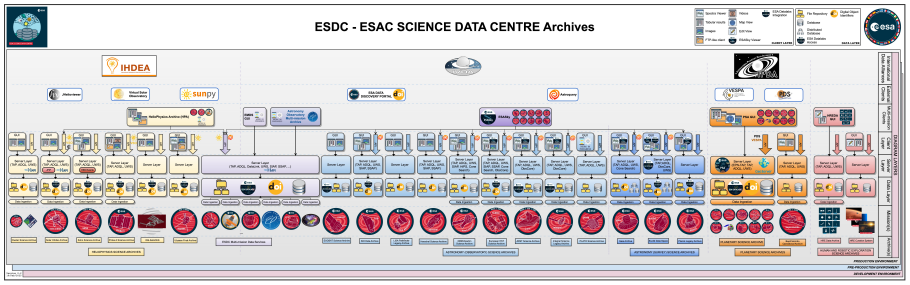


Figure 4. ESDC archives architecture diagram where each box in the figure represents one of the missions ESDC is supporting across astronomy, planetary science, heliophysics and human and robotic exploration domains, all following a consistent three-tier client, server and data layer architecture. The figure also shows the EMDS platform in the middle, which provides a shared infrastructure — common storage, servers, and standardized protocols like TAP, ADQL, etc, so multiple missions can plug into a single platform. It also supports different types of user interfaces, from mission-specific GUIs to sub-area portals —all connected to the same underlying data services.

5.3. Interoperability and standards

Interoperability is a cornerstone of ESDC's technical ecosystem. Standardised server-layer interfaces—such as TAP, ADQL, HAPI, and related protocols—enable consistent querying and retrieval across missions.

These standards support cross-mission analyses, international collaboration, and integration with external tools and archives. Their consistent adoption ensures that ESA data can be used seamlessly within the global scientific infrastructure.

5.4. Evolving for the Future While Preserving the Past

5.4.1. Multi-mission Archives

Long-term sustainability represents one of the most significant challenges for space science archives. While archive systems must evolve to remain technically relevant, they must also preserve uninterrupted access to decades of accumulated scientific data. At ESDC, archive infrastructures have evolved continuously for more than three decades, and this evolution must continue in order to remain maintainable, secure, and usable.

A key sustainability challenge arises from the growing imbalance between active and legacy missions. Projections indicate that by the late 2020s, the number of missions in legacy phase will exceed those in active operations. Legacy missions often no

longer benefit from dedicated funding or personnel, despite the fact that their data remain scientifically valuable for many decades and they continue to lead to highly cited scientific papers (De Marchi & Parmar 2024). This shift fundamentally changes the operational model of archive centres, placing increasing pressure on shared infrastructure and long-term maintenance strategies.

To address this challenge, ESDC is progressively transitioning from isolated, mission specific archives toward multi-mission archives built on shared standards, services, and infrastructure. Examples of this approach include the Planetary Science Archive (PSA: `https://psa.esa.int/`), the Heliophysics Archive (HPA: `https://hpa.esa.int/`), and the Human and Robotic Exploration Data Archive (HREDA: `https://hreda.esac.esa.int/`). These archives consolidate multiple missions within a unified technical framework, reducing operational costs, simplifying maintenance, and enabling immediate interoperability across missions.

The EMDS platform detailed above underpins this transition. EMDS provides shared storage, compute, database services, and standardised access interfaces that allow missions to be integrated early in their lifecycle and remain supported efficiently as they transition into legacy phases. By adopting this model, ESDC improves long-term sustainability while ensuring continued accessibility of mission data.

Importantly, this multi-mission strategy is not limited to planetary science or heliophysics. ESDC is actively exploring the extension of this approach to astronomy, with the goal of bringing missions such as Hubble Space Telescope (HST), James Webb Space Telescope (JWST), and XMM-Newton into a more unified, interoperable environment. Such consolidation enables cross-mission discovery, analysis, and reuse while maintaining mission-specific identities where required.

5.4.2. Planned Technology Evolution

Long-term preservation is not solely a matter of funding or infrastructure capacity; it is equally dependent on systematic technology evolution. Software frameworks, hardware platforms, data formats, and security requirements evolve continuously, and archive systems must adapt accordingly to remain operational over decades.

At ESDC, technology evolution is addressed through a structured and proactive approach. Approximately every five to seven years, archive systems are reviewed and modernised to ensure continued accessibility, maintainability, and security. This process allows ESDC to adopt modern technologies while preserving stable interfaces for users and external services.

Examples of such evolution include the migration from Java applets to modern web frameworks such as Angular for user interfaces, the replacement of bespoke in-house services with open-source, standards-based solutions, and the transition from standalone databases to distributed database systems such as Greenplum. Each transition is designed to improve scalability, maintainability, and interoperability without disrupting access to historical data.

The guiding principle behind this approach is continuity. As technology evolves, archives must evolve with it, but never at the expense of breaking access to decades of accumulated scientific data. This balance between innovation and stability is a defining characteristic of ESDC's data lifecycle management strategy.

5.4.3. Cloud Adoption Considerations

As archive infrastructures continue to scale, the potential use of cloud-based technologies has become an increasingly common topic of discussion. At ESDC, the question is not whether cloud technologies offer benefits, but how and under what conditions they could be adopted sustainably.

Migrating archives and science platforms such as ESA Datalabs to the cloud requires careful consideration. ESDC manages both public and proprietary data, making data access control, identity management, and the colocation of storage and computing critical factors. Cloud adoption is therefore not merely a technical decision, but one that also involves expertise development, governance models, and long-term cost analysis.

The potential benefits of cloud technologies—including elasticity, reliability, and rapid scalability—are attractive as data volumes and user demands continue to grow. However, challenges such as data egress costs pose risks to the sustainability of open access archives, particularly when serving large international user communities.

As a European institution, ESA also considers alignment with European cloud providers and policies. These aspects are currently being evaluated within the ESA Datalabs and Archives Cloud Study Project (EDACS), which will assess the technical, financial, and policy implications of potential cloud adoption. This structured evaluation ensures that any future transition is informed, sustainable, and aligned with ESA's long-term data stewardship responsibilities.

5.4.4. Managing Unplanned Technology Evolution

The evolution of archive technologies is not always predictable. For several years, ESDC relied on Greenplum, a distributed PostgreSQL-based database system, to support large-scale analytical workloads across multiple missions. Greenplum offered parallel query execution, horizontal scalability, and strong SQL compatibility, making it well suited to archive use cases.

In 2024, the discontinuation of the open-source version of Greenplum introduced long-term sustainability concerns. Although existing deployments remained functional, the lack of future community-driven development prompted an ESA-wide assessment of alternatives.

Crucially, ESDC's modular three-tier architecture ensured that this technology shift was contained within the data layer. Client interfaces, APIs, and interoperability services continued to operate unchanged. This experience demonstrated that architectural resilience is achieved not by avoiding change, but by designing systems that can adapt to it.

6. Social Dimensions of Data Lifecycle Management

While scalability, architecture, interoperability, and technology evolution form the technical foundations of data lifecycle management at ESDC, technology alone is insufficient to sustain a long-term scientific archive. Equally important are the social dimensions: the people, policies, governance, and collaborations that enable technical systems to function effectively over decades.

The social dimension encompasses the organisational and governance mechanisms that support data stewardship beyond purely technical concerns. These include cross-

agency coordination, data access and citation policies, knowledge continuity across mission lifecycles, and sustained engagement with the scientific community.

Cross-agency collaboration is central to ESDC's archive strategy. ESA actively contributes to international coordination and standardisation bodies such as the International Virtual Observatory Alliance (IVOA), the International Planetary Data Alliance (IPDA), and the International Heliophysics Data Environment Alliance (IHDEA), enabling alignment of data models and interoperability standards across agencies and scientific domains (Quinn et al. 2004; Hare et al. 2017). In parallel, ESDC works closely with major space agencies, including National Aeronautics and Space Administration (NASA), Japan Aerospace Exploration Agency (JAXA), and Chinese Academy of Sciences (CAS), and with key data centres such as Centre de Données astronomiques de Strasbourg (CDS) and Space Telescope Science Institute/Mikulski Archive for Space Telescopes (STScI/MAST). Additional collaborations with the LSST Interdisciplinary Network for Collaboration and Computing (LINCC) Frameworks team for the Vera Rubin Observatory, EU-funded initiatives such as Scalable Processing of Astrophysical Data in the Cloud for fUture Science (SPACIOUS), and mission-specific partner data centres (e.g. DZA for Gaia and Infrared Processing and Analysis Center IPAC for Euclid) ensure distributed expertise and shared operational responsibility throughout the data lifecycle.

Clear policies and governance frameworks underpin responsible data access, citation, and reuse. ESA defines transparent rules governing proprietary periods, licensing, and attribution, supporting scientific integrity, reproducibility, and alignment with open science and FAIR principles.

As missions transition into legacy phases, preserving institutional knowledge becomes critical. At ESDC, this is addressed through stable teams, comprehensive documentation, and shared platforms such as the EMDS, which reduce fragmentation and preserve expertise across mission lifetimes.

Finally, ESDC actively supports and engages the scientific community through intuitive interfaces, programmatic access, responsive helpdesks, and structured feedback channels such as the Archives Users Groups, which are advisory boards external to ESA advising ESDC development . Open, standards-based protocols, including TAP, ADQL, and HAPI, enable interoperable access across international infrastructures and maximise long-term data reuse (Dowler et al. 2019).

7. Conclusions

Data lifecycle management is fundamental to ensuring that space science data remain usable, discoverable, and scientifically valuable long after mission operations have ended. At ESA's ESAC Science Data Centre, this responsibility spans the full lifecycle of mission data and must address both rapidly growing data volumes and the increasing dominance of legacy missions.

This paper shows that sustainable archives depend on the combined strength of technical and social dimensions. Scalable architectures, shared platforms, and interoperability provide the technical foundation for managing petabyte-scale, multi-mission datasets. Equally important are governance, international collaboration, knowledge continuity, and active engagement with the scientific community, which ensure that these systems function effectively in practice and remain trusted over time.

ESDC archives have evolved from static repositories into active enablers of science, offering discovery, access, and analysis services that support cross mission and interdisciplinary research. Initiatives such as EMDS and multi-mission archives demonstrate how shared infrastructure can improve sustainability while reducing long-term maintenance costs.

Finally, long-term preservation requires planning for change. Technology evolution and external dependencies make adaptability a critical design principle. The experiences described here illustrate how modular, flexible architectures can absorb unplanned technological shifts without disrupting access to accumulated data. These lessons are relevant beyond ESA and may inform the design of future astronomical and interdisciplinary data infrastructures.

References

Arviset, C., et al. 2026, in ADASS XXXV, edited by TBD (San Francisco: ASP), vol. TBD of ASP Conf. Ser., TBD

De Marchi, G., & Parmar, A. N. 2024, arXiv e-prints, arXiv:2402.12818. eprint: 2402.12818

Dowler, P., et al. 2019, Table Access Protocol Version 1.1, IVOA Recommendation 27 September 2019

Hare, T. M., et al. 2017, Earth and Space Science, 4, 1

Quinn, P. J., et al. 2004, in Optimizing Scientific Return for Astronomy through Information Technologies, edited by P. J. Quinn, & A. Bridger, vol. 5493 of Society of Photo-Optical Instrumentation Engineers (SPIE) Conference Series, 137

Wilkinson, M. D., et al. 2016, Scientific Data, 3, 160018

Astronomical Data Analysis Software and Systems XXXV
*ASP Conference Series, Vol. **VOL** Compiled: Tuesday 30th June, 2026@ 01:30*
K. Polsterer, D. Bohlender, S. Gwyn, A. Hincks and P. Teuben, eds.
*© **YEAR** Astronomical Society of the Pacific*

ZZZDEBUG!020.tex
Jiang,H.|textbf
Rahman,M.
Scora,J.
Withers,T.

Generating Astronomical Multiband Images using the Hue-Saturation-Value Colour Space

Hansen Jiang, Mubdi Rahman, Jennifer Scora, Tai Withers

Sidrat Research, Toronto, Ontario, Canada; hansen@sidratresearch.com

Abstract. Image generation is an important step in the modern astronomy data analysis workflow. It provides quick-look diagnostics on raw data or during the data reduction stages, enabling visual identification or classification of sources and features, and the presentation of the data to the larger scientific community. Traditionally, these images are created from stacking three (or more) scaled single band images as colour channels, in a standard RGB (sRGB) image. This technique belies the complexities of colour science and perception, where even the simplest of images require complex transformations between the input gamut from the camera sensors to the final output sRGB. In this paper, we present a method of combining any number of astronomical single band images into sRGB images using the Hue-Saturation-Value (HSV) colour space. This method produces output images that are better representative of noise properties, structures, and features of the underlying data, makes visualization choices more explicit, and produces more visually appealing images.

1. Introduction

The multi-band image is a cornerstone of astronomical data visualization. It provides diagnostic insights during collection, reveals features and structure during analysis, and represents a milestone for a project for the wider astronomical community. The familiar method of generating these images, however, is based on a technique using a colour model that limits the scope of visualization, and is wrong for the task - namely, the RGB colour model. By processing multi-band data in a different colour model, data can be depicted with scientifically meaningful choices that are more suited for human interpretation.

2. Choosing A New Colour Model

2.1. The RGB Colour Model

The RGB colour model is an additive colour model based on three primary colours - red, green, and blue. It was designed to mimic human vision and provide a framework to efficiently reproduce colours for electronic display. However, it assumes these red, green, and blue axes are independent, as with RGB sub-pixel LEDs in an electronic display. Human vision is much more complex, with rod and cone cells determining brightness, and cone cells determining colour, but being interdependent with highly overlapping, correlated, and inseparable responses. Consequently, human perception of colour is not described via these RGB axes (Rhyne 2017).

In this model, multi-band images are generated by using three broadband-filtered observations as input for each of the primary red, green, and blue channels. In doing so, brightness information is used as colour information, meaning transformation in either axis is inseparable from the other. And while merging these channels produces an image, the final visualization is often a result of "magic numbers" used to adjust the output of each of the three channels, for example to reduce red tint caused by a brighter long-wavelength observation. Thus, this method and the use of these magic numbers separate visualization choices from the data itself.

2.2. The HSV Colour Model

In comparison, the HSV model is a colour model based on axes closer to human perception of colour through rods and inseparable cones - hue, saturation, and value. These axes are trivially transformable to RGB coordinates, putting chromatic decisions up front while still optimizing for computer display. The inputs for each channel are not a one-to-one with the data as in the RGB model, but rather a calculation involving data across filters. This proceeding presents one such method of calculation for each of the HSV channels.

The hue channel defines the "pure colour" at any given pixel within the image, so its calculation is set to the average colour across all bands. With angular hue values $0° \leq \alpha \leq 360°$, the average is found on a per-pixel basis via the weighted circular mean. The weight is defined as the band's ratio to the global maximum, at some percentile q_1. The saturation channel defines the "vividness" of the colour at any given pixel, which in this method is used to represent a measure of difference in information across bands. For each pixel, the maximum peak to peak distance is calculated, then weighted by the band's ratio to the global maximum at some percentile q_2, similarly to the hue channel. The brightness channel is calculated as normally would be - that is, as a sum of the intensity information from each filter, log-normalized, and scaled to the full dynamic range.

Combining these channels produces the HSV image as presented in Figure 1.

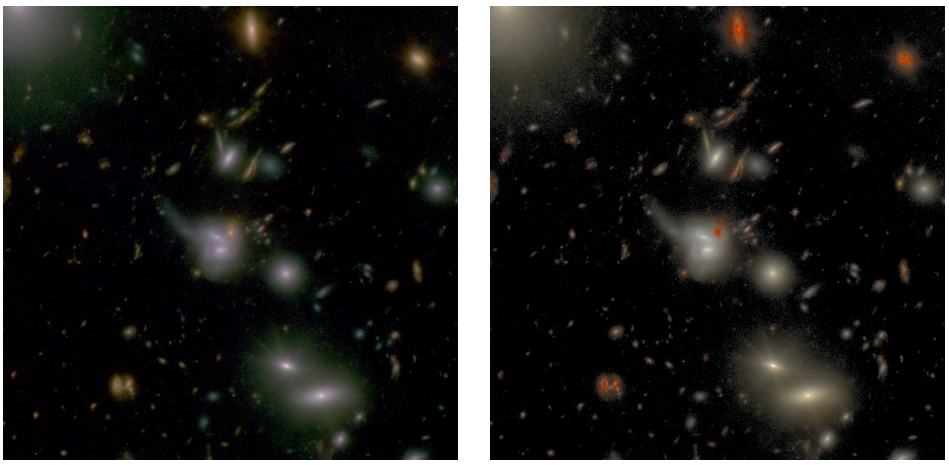


Figure 1. Comparing the visualization of a small sub-field in the Abell 2744 cluster, as visualized in the RGB colour model [left] and the HSV colour model [right].

3. Making Choices in Data Visualization

Using the same observation data, drastically different visualizations can be made by choice of colour model. Notably, the ability to represent data based on scientifically meaningful decisions becomes evident when working in the HSV model.

3.1. Noise

In the RGB colour model, noise from each band is depicted by its corresponding channel's colour. This can create the illusion of structure due to wide variations of differences between bands, corresponding to significant hue and saturation variations. By contrast in the HSV colour model, the noise is most evident in the brightness, and so there are minimal artifacts from the difference between bands, as can be compared in Figure 2.

3.2. Structure

In RGB, object structures and features are limited to the output from the noisiest band, especially if the channels are adjusted to account for the noisiness. Comparatively in HSV, brightness is based on the total signal-to-noise, rather than the strength of any one band's flux. This results in object structures appearing sharper, with finer and dimmer features becoming visible, not being obscured by colour information, and having higher contrast, as can be seen in Figure 2.

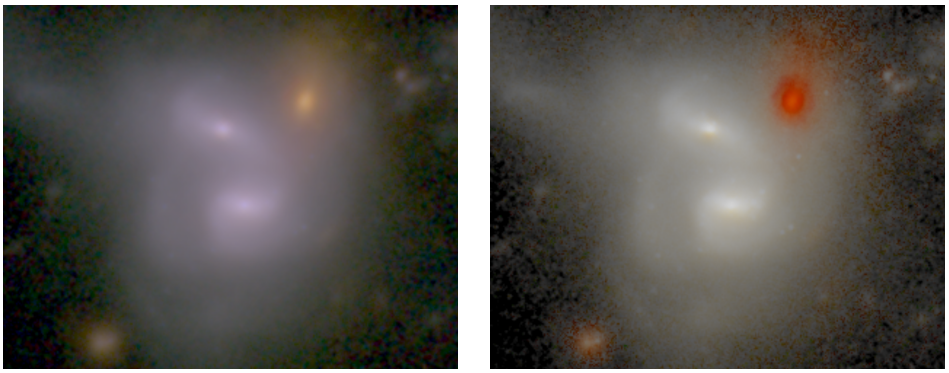


Figure 2. Comparing the depiction of noise and structure, as visualized in RGB [left] and HSV [right].

3.3. White Balance

With each filter set to a particular hue, the "neutral point" of the image can be set, analogous to white balancing an image. This method can be used to contrast objects of different redshifts.

3.4. Additional Bands

In RGB, making an image from any number of bands other than three requires weighted distribution across channels, or the use of other pre-processing methods. For the sake of comparison, the HSV image presented in Figure 1 uses the same three bands as the

RGB image. However, in HSV the number of bands used is trivially variable, where the addition of observations provides additional structure information and reduces the effects of noise, as can be seen in Figure 3.

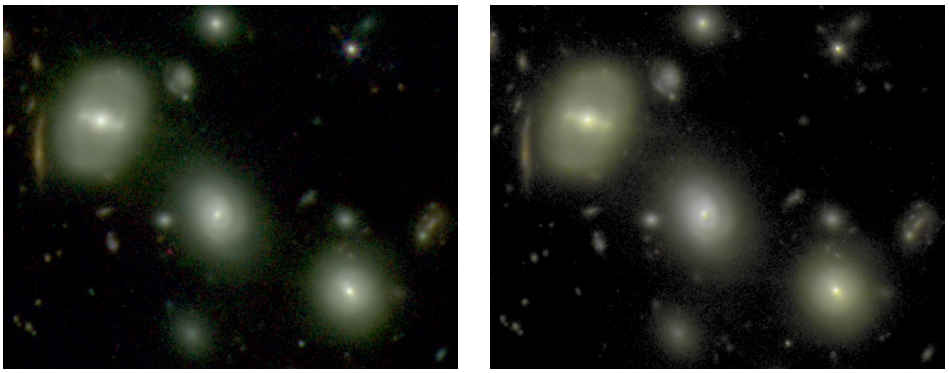


Figure 3. Comparing the effects of using 3 bands in RGB [left] and 8 bands in HSV [right].

3.5. Depicting Difference

With colour and saturation depicting the difference of information across the filters, an image's colour can be adjusted based on what features are prioritized.

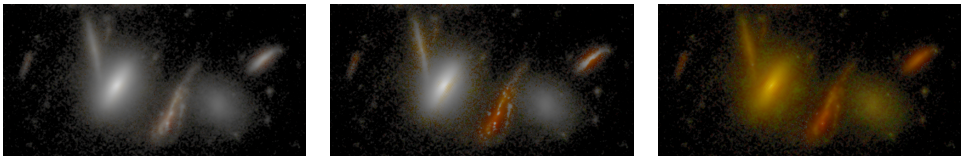


Figure 4. Adjusting saturation thresholds to highlight different features in objects, from left to right - increased thresholds, selectively saturating features which are highly different between bands, and decreased thresholds.

4. Next Steps

This HSV colour multi-band image generation method is being developed into a library. Feel free to reach out to the authors for more information.

Acknowledgments. Some of the data products presented herein were retrieved from the Dawn JWST Archive (DJA; Brammer 2023; Valentino et al. 2023). DJA is an initiative of the Cosmic Dawn Center (DAWN), which is funded by the Danish National Research Foundation under grant DNRF140.

References

Brammer, G. 2023, grizli. URL https://doi.org/10.5281/zenodo.8370018
Rhyne, T.-M. 2017, Applying Color Theory to Digital Media and Visualization (CRC Press)
Valentino, F., et al. 2023, ApJ, 947, 20. eprint: 2302.10936

Astronomical Data Analysis Software and Systems XXXV
*ASP Conference Series, Vol. **VOL** Compiled: Tuesday 30th June, 2026@ 01:30*
K. Polsterer, D. Bohlender, S. Gwyn, A. Hincks and P. Teuben, eds.
*© **YEAR** Astronomical Society of the Pacific*

ZZZDEBUG!021.tex
Rahman, M.|textbf
journals

How to Write a Software Paper for the AAS Journals

Mubdi Rahman[1,2]

[1]*Sidrat Research, Toronto, Ontario, Canada;* mubdi@sidratresearch.com

[2]*American Astronomical Society Journals, Washington DC, USA*

Abstract. Software tools and the algorithms underlying them have become critical to the advancement of astronomical research. The contribution of those who develop astronomical software can and should be directly linked to the discoveries made using these tools. The American Astronomical Society Journals, including the Astrophysical Journal and the Astronomical Journal, explicitly welcome articles whose purpose is to describe the design and function of software with relevance to research in astronomy and astrophysics. In this talk, I described what a software paper is, some best practices on how to write one, and the editorial processes that happen behind the scenes to get your manuscript from an idea to a published paper. I further discussed a few features of publishing software papers within the AAS journals, including Living Papers and our partnership with the Journal of Open Source Software.

1. What is a Software Paper?

Software is a key driver of scientific discovery in astronomy. Papers that describe research software are important to the progress of the field, providing a description of the scientific and technical decisions made in the design of the package, a means to interrogate how the underlying system works, and appropriate attribution to those involved in the creation of the tool. Consequently, the American Astronomical Society (AAS) Journals have over the past decade introduced an initiative to accept and publish **Software Papers** in its journals (Vishniac & Lintott 2016) through the adoption of its Software Paper Policy[1].

A software paper is a peer-reviewed paper appearing in one of the AAS Journals (e.g., The Astronomical Journal, The Astrophysical Journal, The Astrophysical Journal Letters, The Astrophysical Journal Supplement Series, or the Planetary Science Journal) that describes the code/software that has been developed in aid of scientific research. As per the journals' policy on software, the paper itself does not need to include specific research results, though the inclusion of some may be useful to describe the application of the software.

While we encourage it, the software being described in the paper does not need to be open source. It merely needs to be important to published research.

While a software paper does not have a minimum length or prescribed structure, software papers should not be the following:

[1]https://journals.aas.org/policy-statement-on-software/

- Software papers are not the documentation of the software.
- Software papers are not guarantees of continued support or development of the software.
- Software papers are not a requirement to publish ones code.
- Software papers are not a guarantee of the quality or importance of the code.

A software paper should provide a fellow astronomer with sufficient detail to understand what what science can be conducted with the software package and how the software has been implemented.

2. Why Should you Publish a Software Paper?

There may be many motivations to publish a software paper, these may include:

Attribution and Credit: Software papers ensure that those involved in the design and development of the code receive the appropriate academic attribution.

Peer Scientific Validation: Validate the scientific choices and technical implementation of these choices through the peer-review process.

Promotion: Promote the use of a specific software package for use by the broader community, including across sub-fields of astronomy.

Community Development: Identify interested scientists who may want to contribute to or help maintain the software you have developed.

Future Ideation: Inspire other researchers to use and extend the software and scientific methods that you are developing.

3. Structure of a Software Paper

Largely, the target audience for a software paper is other astronomers, not, for instance, computer scientists. While the AAS Journals do not prescribe a specific format or structure for a software paper, we recommend addressing the following questions:

- What is the challenge that the software seeks to address?
- How is the software implemented technically?
- What scientific choices have been made in the software?
- How does one use the software/code?
- What's an example use case of the software?

Addressing these questions will enable the astronomical community to gain the greatest insight into the software package, maximizing the relevance of the paper.

4. Value-Added Features through the AAS Journals

There are unique aspects of software papers that are distinct from other publishable manuscripts that appear in the journals. For instance, research software is often ever-evolving. Consequently, we have introduced Living articles (Vishniac & Lintott 2018). Living articles are a feature that allows authors to update and add to a paper as the underlying code/software evolves. Each version of the paper goes through peer-review and can be specifically cited. Further, new co-authors may be added to the paper.

Further, the AAS Journals has a partnership with the Journal of Open Source Software (JOSS) where the software code can be directly peer-reviewed and published alongside the software paper (Vishniac & Lintott 2018). JOSS will validate, review, and publish your software to ensure that they meet a "gold-standard" of quality. Further, the AAS Journals provides a nominal fee for this service to JOSS on behalf of the author.

5. Conclusions

Software papers are becoming an increasingly important component of the astronomical software ecosystem, providing astronomers with a means to communicate the scientific insight and design of the software tools used to conduct research. The Journals of the American Astronomical Society are the second-to-none venue to publish these papers a specific policy in place to around their submission and peer-review, an entire editorial corridor dedicated to their handling, and value-added features to expand the impact of software papers published within. We encourage the submission of software papers to the AAS Journals as you develop your next pieces of research software.

References

Vishniac, E. T., & Lintott, C. 2016, AJ, 151, 21

Vishniac, E. T., & Lintott, C. 2018, The Astrophysical Journal, 869, 156. URL `https://doi.org/10.3847/1538-4357/aaf876`

Vishniac, E. T., & Lintott, C. 2018, ApJ, 868, 78

Astronomical Data Analysis Software and Systems XXXV
*ASP Conference Series, Vol. **VOL** Compiled: Tuesday 30th June, 2026@ 01:30*
K. Polsterer, D. Bohlender, S. Gwyn, A. Hincks and P. Teuben, eds.
*© **YEAR** Astronomical Society of the Pacific*

ZZZDEBUG!022.tex
Scora, J.|textbf
Rahman, M.
Jiang, H.
Kemp, J.
Iyer, K.
Mowla, L.
Withers, T.
observatories!space-based!
GLASS! ascl:1806.009
GALAXY! ascl:1904.002

A Platform to Unify Community-Generated JWST Legacy Data Products

Jennifer Scora,[1] Mubdi Rahman,[1] Hansen Jiang,[1] Jonathan Kemp,[2] Kartheik Iyer,[3] Lamiya Mowla,[2] and Tai Withers[1]

[1]*Sidrat Research, Toronto, Ontario, Canada;* jscora@sidratresearch.com

[2]*Wellesley College, Wellesley, Massachusetts, USA*

[3]*Columbia University, New York, New York, USA*

Abstract. The James Webb Space Telescope is producing a firehose of extragalactic imaging data through its diversity of legacy programs. Community organized initiatives, such as the Dawn JWST Archive, have come to fill the gap between archive products to uniformly-reduced data that enable large-scale exploration and analysis. These programs are catalyzing further initiatives to generate value-added catalogs of inferred parameters, such as those from SED fitting, morphology extraction, and machine learning algorithms. Using and comparing these catalogues will become increasingly challenging as more data is made available and new parameters are created.

Through the J-HIVE initiative, we have created a platform to generate purpose-specific catalogs that combine the community generated data products in a versionable and traceable manner. The platform also generates schema to enable access to all of this data in a discoverable and documented manner. The Python/JSON based system can be easily modified to add in new catalogues or datasets as they are available. It also performs simple transformations and filters on the data, which can be added to as needed. The output catalogue can then be easily pushed to visualization mechanisms for exploration and analysis. In this talk, we will present how the platform was developed and the benefits it provides to researchers using JWST data now and in the future.

1. Introduction

The James Webb Space telescope (JWST) is producing a firehose of extragalactic imaging data through its diversity of legacy programs, such as the CAnadian NIRISS Unbiased Cluster Survey (CANUCS) (Sarrouh et al. 2026), Ultradeep NIRSpec and NIRCam ObserVations before the Epoch of Reionization (UNCOVER) (Bezanson et al. 2024) and GLASS (Treu et al. 2022). However, with such a wide variety of legacy programs, the data is not always easily compared or compiled. It has fallen upon the community to gather the data from all of these different legacy surveys, and reduce them in a consistent way so that they can be treated as one large dataset. Community organized initiatives, such as the Dawn JWST Archive (DJA), have provided this service, uniformly reducing the data and making it available (Valentino et al. 2023; Brammer 2023).

The goal of JWST's Homogeneous Information-Rich Visualization of Extragalactic Datasets (J-HIVE) is to combine these catalogues of observed magnitudes with cat-

alogues of derived data parameters from methods such as SED fitting (Iyer & Gawiser 2017), morphology (Mowla et al. 2024), extraction, and machine learning algorithms, and to present that data in an interactive visualization that enables researchers to explore the data. [1] In this paper, we describe a new framework that we created for the purpose of combining JWST galaxy data from the DJA and derived data parameters into one comprehensive, versioned, easy to access catalogue.[2]

2. The Framework

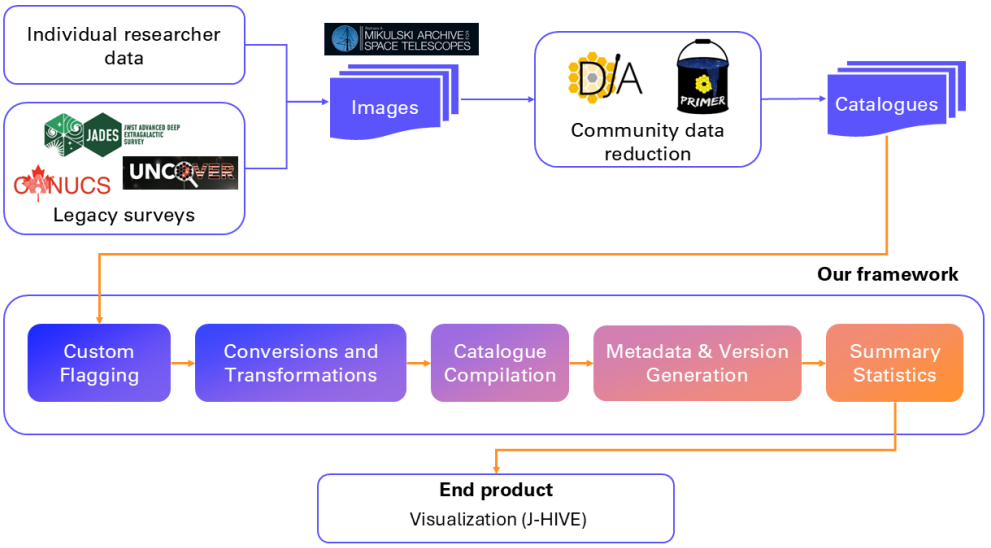


Figure 1. This is the main workflow that we built the framework for, which produces output catalogues, metadata, and summary statistics that are used by the J-HIVE visualization.

We designed the framework to be modular, so it could be easily adapted to different input catalogues and different user needs. Figure 1 shows the workflow currently in use to prepare the data for the J-HIVE visualization, which makes very linear use of the modules, but most of these modules could be used on their own, or in other configurations. Below we describe each module in detail.

2.1. Metadata and version generation

The configuration files are a key part of the framework that enable the whole thing to function. Each file contains important organizational information such as the paths to all the catalogues (i.e. DJA data, morphological parameters, SED fitting parameters), and lists of columns from each catalogue that need to end up in the final combined catalogue. It also includes the path to the output directory and the version number for

[1] Take a look at the J-HIVE visualization here: `https://j-hive.org/`

[2] The code for the framework can be found here: `https://github.com/j-hive/jhive-previz`

that specific configuration. Given a specific version of an output catalogue, one should be easily able to identify which catalogues were used to create that output catalogue. Essentially, the goal of these files is to make it easy to track the data provenance of the output catalogues.

In addition to the configuration files described above, there are also metadata files for each specific catalogue type. These files includes information about the column names, units, and so on of all the columns that could be in such a file, and are used by most modules to identify relevant information about the data columns. A similar `JSON` metadata file is created to go with the output catalogue, for use by the visualization system.

These files already contain much of the useful data that a user may want to know about a catalogue, but in difficult to read formats. To make things easier for the user, we created a module that turns these files into `csv` tables that are automatically uploaded to the documentation for the project. [3] This both makes creating and editing the documentation easier for the developers.

2.2. Custom flags

The flagging module cuts the data based on certain defined flags. It outputs a `FITS` file with the flags for a specific catalogue, with a flag for each column of an object and then a cumulative flag for each object. An example use case would be to cut for "good" data, where the signal-to-noise ratio is above a certain threshold for a certain number of filters. However, the module can be customized as desired, to allow for users to create their own cuts.

2.3. Conversions and transformations

This module is an important utility for any framework, as it is often necessary to transform the input data in some way. It can perform arbitrary transformations of the data to ensure the data in the output catalogue is in the desired formats. One example is the conversion of a column from its input units to a different unit in the output catalogue. This can be performed by tracking the input format and the desired output format in the metadata files for the catalogue.

2.4. Catalogue compilation

A core functionality of the framework is combining catalogues together. The code uses the columns listed in the configuration files to get the subset of desired data from each input catalog, then uses `pandas` to join each of these subsets of catalogues together on the object ID column, one at a time. Once these catalogues are all joined together, the output catalogue (or catalogues) are written to their own folder.

2.5. Summary statistics

The summary statistics module computes column-wise summarized quantities for the output catalogues. An example of these are histograms of the distribution of values in each column, or the minimum and maximum value of each column. Both of these give

[3] You can see the tables here: `https://j-hive.github.io/jhive-docs/previz/dataschema.html`

the user a sense of the data contained in the catalogue, and facilitate further plotting and so on, which are the goals of the summary statistics module.

3. Where do we go from here?

We have created a framework that combines observed JWST extragalactic data with derived parameters, making it easier for scientists to perform large data projects on these data. The framework itself is designed to be flexible, traceable, and both easy to use and easy to maintain. Currently, it is being used on the JWST extragalactic legacy data products as described in this article, but it could easily be adapted for use on other data sets.

References

Bezanson, R., Labbe, I., Whitaker, K. E., Leja, J., Price, S. H., Franx, M., Brammer, G., Marchesini, D., Zitrin, A., Wang, B., Weaver, J. R., Furtak, L. J., Atek, H., Coe, D., Cutler, S. E., Dayal, P., van Dokkum, P., Feldmann, R., Förster Schreiber, N. M., Fujimoto, S., Geha, M., Glazebrook, K., de Graaff, A., Greene, J. E., Juneau, S., Kassin, S., Kriek, M., Khullar, G., Maseda, M., Mowla, L. A., Muzzin, A., Nanayakkara, T., Nelson, E. J., Oesch, P. A., Pacifici, C., Pan, R., Papovich, C., Setton, D. J., Shapley, A. E., Smit, R., Stefanon, M., Taylor, E. N., & Williams, C. C. 2024, ApJ, 974, 92. eprint: 2212.04026

Brammer, G. 2023, grizli. URL `https://doi.org/10.5281/zenodo.8370018`

Iyer, K., & Gawiser, E. 2017, ApJ, 838, 127. eprint: 1702.04371

Mowla, L., Iyer, K., Asada, Y., Desprez, G., Tan, V. Y. Y., Martis, N., Sarrouh, G., Strait, V., Abraham, R., Bradač, M., Brammer, G., Muzzin, A., Pacifici, C., Ravindranath, S., Sawicki, M., Willott, C., Estrada-Carpenter, V., Jahan, N., Noirot, G., Matharu, J., Rihtaršič, G., & Zabl, J. 2024, Nat, 636, 332. eprint: 2402.08696

Sarrouh, G. T. E., Asada, Y., Martis, N. S., Willott, C. J., Iyer, K. G., Noirot, G., Muzzin, A., Sawicki, M., Brammer, G., Desprez, G., Rihtaršič, G., Zabl, J., Abraham, R., Bradač, M., Doyon, R., Antwi-Danso, J., Berek, S., Brown, W., Estrada-Carpenter, V., Favaro, J., Felicioni, G., Forrest, B., Gaspar, G., Gould, K. M. L., Gledhill, R., Harshan, A., Jahan, N., Jagga, N., Judež, J., Marchesini, D., Markov, V., Matharu, J., MacFarland, S., Merchant, M., Mérida, R. M., Mowla, L., Myers, K., Omori, K. C., Pacifici, C., Ravindranath, S., Robbins, L., Strait, V., Sok, V., Tan, V. Y. Y., Tripodi, R., Wilson, G., & Withers, S. 2026, ApJS, 282, 3. eprint: 2506.21685

Treu, T., Roberts-Borsani, G., Bradac, M., Brammer, G., Fontana, A., Henry, A., Mason, C., Morishita, T., Pentericci, L., Wang, X., Acebron, A., Bagley, M., Bergamini, P., Belfiori, D., Bonchi, A., Boyett, K., Boutsia, K., Calabró, A., Caminha, G. B., Castellano, M., Dressler, A., Glazebrook, K., Grillo, C., Jacobs, C., Jones, T., Kelly, P. L., Leethochawalit, N., Malkan, M. A., Marchesini, D., Mascia, S., Mercurio, A., Merlin, E., Nanayakkara, T., Nonino, M., Paris, D., Poggianti, B., Rosati, P., Santini, P., Scarlata, C., Shipley, H. V., Strait, V., Trenti, M., Tubthong, C., Vanzella, E., Vulcani, B., & Yang, L. 2022, ApJ, 935, 110. eprint: 2206.07978

Valentino, F., Brammer, G., Gould, K. M. L., Kokorev, V., Fujimoto, S., Jespersen, C. K., Vijayan, A. P., Weaver, J. R., Ito, K., Tanaka, M., Ilbert, O., Magdis, G. E., Whitaker, K. E., Faisst, A. L., Gallazzi, A., Gillman, S., Giménez-Arteaga, C., Gómez-Guijarro, C., Kubo, M., Heintz, K. E., Hirschmann, M., Oesch, P., Onodera, M., Rizzo, F., Lee, M., Strait, V., & Toft, S. 2023, ApJ, 947, 20. eprint: 2302.10936

Astronomical Data Analysis Software and Systems XXXV
*ASP Conference Series, Vol. **VOL** Compiled: Tuesday 30th June, 2026@ 01:30*
K. Polsterer, D. Bohlender, S. Gwyn, A. Hincks and P. Teuben, eds.
*© **YEAR** Astronomical Society of the Pacific*

ZZZDEBUG!025.tex
Muraleedharan, D.|textbf
Hirzberger, J.
Staub, J.
Michalik, H.
Solanki, S. K.

Real-time onboard front-end data processing pipelines for the Photospheric Magnetic field Imager on ESA's Vigil mission

Deepa Muraleedharan,[1,2] Johann Hirzberger,[1] Jan Staub,[1] Harald Michalik,[2] and Sami K. Solanki[1]

[1]*Max Planck Institute for Solar System Research, Göttingen, Germany;* muraleedharan@mps.mpg.de

[2]*Technical University Braunschweig, Braunschweig, Germany*

Abstract. The ESA *Vigil* mission, positioned at the Sun-Earth L5 Lagrange point, is designed to provide continuous solar monitoring for space-weather forecasting and heliophysics research. The mission's Photospheric Magnetic field Imager (PMI) acquires full-disk solar images generating raw data rates far exceeding the available telemetry bandwidth. To meet stringent latency and telemetry constraints, PMI implements real-time onboard data reduction using a Field-Programmable Gate Array (FPGA)-based processing system that converts high-volume filtergrams into calibrated, demodulated, and compressed science products. This paper presents the front-end stages of the onboard pipeline - the Accumulator and Preprocessing modules, which transform raw filtergrams into calibrated Stokes vectors through deterministic, parallel execution. The implemented system achieves real-time throughput, bit-accurate correctness, and compliance with the allocated FPGA resource budgets.

1. Introduction

The ESA *Vigil* mission is the first operational space-weather observatory positioned at the Sun-Earth L5 Lagrange point, providing continuous solar observations and early warning of Earth-directed coronal mass ejections. Among its payload instruments, the Photospheric Magnetic field Imager (PMI) (Staub et al. 2020) is a full-disk magnetograph designed to measure the photospheric magnetic field with high cadence and low latency.

Building on the heritage of the SO/PHI instrument aboard *Solar Orbiter* (Solanki et al. 2020), PMI introduces higher onboard autonomy, continuous operation, and enhanced data reduction capabilities. The onboard processing pipeline, implemented on a radiation-tolerant AMD Kintex UltraScale FPGA within the Main Processing Unit (MPU), comprises of four stages: (1) accumulation of multiple exposures, (2) preprocessing with dark/flat-field correction and polarimetric demodulation, (3) inversion of the Radiative Transfer Equation (RTE), and (4) compression and packetization for telemetry (del Toro Iniesta 2003).

The first two stages - accumulation and preprocessing - constitute the front end of the pipeline, converting 24 Gbit of raw filtergrams per dataset into calibrated maps of the Stokes parameters (I, Q, U, V) at six wavelength positions in and near the photospheric Fe I 617.3 nm spectral line (Stokes profiles), which serve as inputs to the

subsequent inversion and compression stages. The complete onboard processing chain achieves an overall data reduction exceeding a factor of 190 while preserving scientific fidelity and meeting the mission's strict cadence and latency constraints. This work extends the implementation presented in Muraleedharan et al. (2025), incorporating additional onboard functionalities and updated performance results.

2. Onboard Data Processing Pipelines and Data Products

The PMI onboard processing chain converts raw filtergrams from the Focal Plane Assembly (FPA) into calibrated science products through three parallel FPGA-based pipelines - Nominal Mode (NM), High-Cadence (HC) Mode, and Raw Mode (RM), each optimized for distinct observation objectives and latency requirements. The Nominal

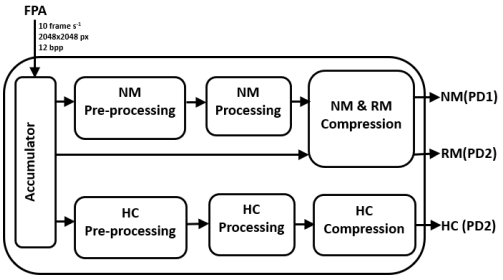


Figure 1. Onboard pipelines and data products. Adapted from Muraleedharan et al. (2025).

pipeline (Priority 1) performs RTE inversion to derive the full disk maps of the photospheric magnetic field strength (B), inclination (γ), azimuth (ϕ), continuum intensity (I_c), and line-of-sight velocity ($v_{\rm LOS}$), delivering calibrated products every 30 min with 20 min latency for near real-time space weather monitoring. The High-Cadence pipeline (Priority 2) provides rapid indicators of solar activity with 1-2 min cadence by computing I_c and $v_{\rm LOS}$ for helioseismic applications and global solar activity studies. The Raw pipeline (Priority 2) provides unprocessed filtergrams at 30-120 min intervals for ground-based calibration and verification. The three pipelines execute in near-real time and in parallel on the FPGA-based Main Processing Unit.

3. Data Processing pipeline implemented in FPGA

3.1. Accumulation

The accumulation stage initiates the onboard processing pipeline, operating directly on raw frames from the 2048 × 2048-pixel FPA (12 bpp, 10 fps), corresponding to an input rate of about 500 Mbit/s sustained in real time. The frames are streamed via a channel-link interface to the Accumulator IP core, which performs continuous pixel-level summation to improve the signal-to-noise ratio.

During each observation sequence, up to twenty modulated filtergrams are acquired for every wavelength and polarization state. The dual-channel readout from the FPA is first reordered to reconstruct the correct pixel geometry, after which the frames are accumulated pixel by pixel in memory. A built-in cosmic-ray detection algorithm

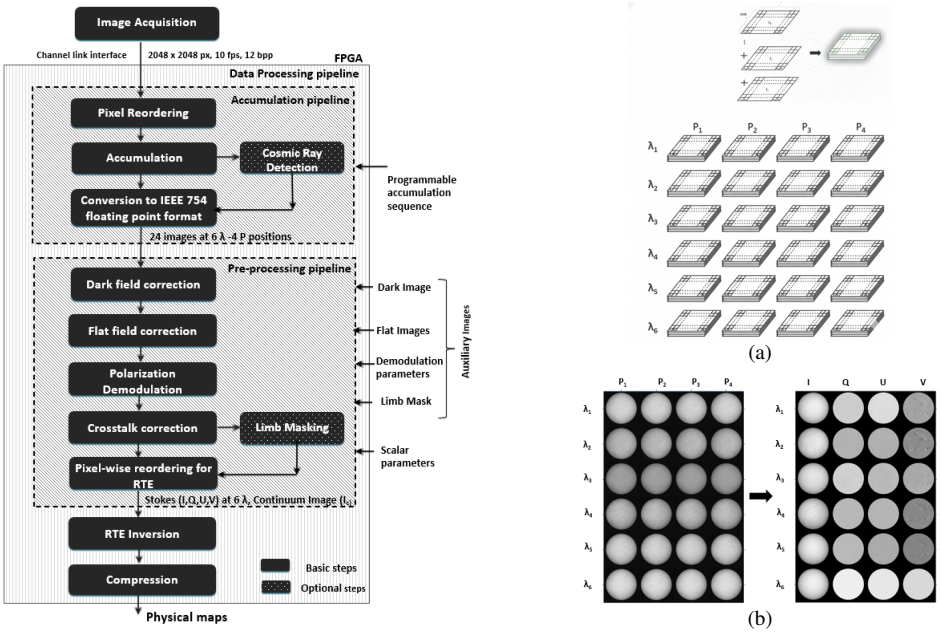


Figure 2. FPGA data processing pipeline (left) and front-end modules: (a) Accumulation, and (b) Demodulation producing Stokes parameters (right). Subfigures (a) and (b) adapted from Muraleedharan et al. (2025), where (b) uses SO/PHI data (2023-04-09T22:00:09).

compares each incoming pixel with the running mean and replaces detected outliers with the mean value, effectively suppressing radiation-induced artefacts before calibration. Sequencing across the wavelength-polarization grid is parameter controlled, allowing accumulation to pause during filter or polarization switching and resume with the next exposure set. The accumulated results are stored as 32-bit floating-point images, forming the input to the subsequent preprocessing stage.

3.2. Nominal Mode Preprocessing

The preprocessing stage operates on the accumulated frames to derive the Stokes profiles, preparing the data for the subsequent RTE inversion. The functions are implemented in a dedicated FPGA IP core and executed entirely in 32-bit IEEE-754 single-precision floating-point format.

This stage performs a sequence of calibration and demodulation steps. First, dark-field correction removes pixel offsets arising from fixed-pattern noise and dark current by subtracting a reference dark frame from each accumulated image. Flat-field correction then compensates for gain variations and optical non-uniformities using wavelength-dependent flat-field maps, which are pre-generated on the ground and include spectral calibration information such as the transmission profile of the order-sorting prefilter across the wavelength scan range.

The corrected filtergrams are subsequently demodulated into the Stokes parameters by multiplying with the pre-characterized inverse of the 4x4 matrix containing the polarimetric modulation scheme. To minimize systematic cross-talk between the Stokes parameters introduced by imperfections in the modulation matrix, scalar correction factors are applied to Q, U, and V according to pre-flight calibration constants.

BAR!conference!ADASS
019
BAR!organisations!ASP
BAR! ascl:1101.010

An optional solar limb-masking step removes regions outside the solar disk to suppress noise propagation and reduce computational overhead. Finally, the resulting Stokes parameter maps, together with the continuum intensity image (I_c) - derived from the Stokes I frame obtained near the Fe I spectral line - are written back to memory, providing input for the Nominal Mode RTE inversion.

4. Results

Building on the results presented in Muraleedharan et al. (2025), the Accumulator IP core now incorporates Cosmic Ray Detection, and both modules include additional onboard functionalities. Updated FPGA resource utilization is given below. Both IP cores meet throughput, timing, and resource constraints. The Accumulator processes a full 2048×2048 dataset (16×6×4 frames) in 27.9 s (13.75 fps), exceeding the 10 fps input rate and completing within the 60 s limit. The Preprocessing core completes Nominal Mode execution in 15.6 s, well below the 120 s requirement. All outputs are bit-accurate with the software reference model, and the implementation meets timing closure at 100 MHz within the FPGA resource budget.

Table 1. FPGA Resource Utilization of the IP Cores

	Accumulator			**Preprocessing**		
Resource	**Used**	**Available**	**(%)**	**Used**	**Available**	**(%)**
LUT	1415	331680	0.60	8374	331680	3.0
FF	2204	663360	0.43	12983	663360	2.10
BRAM	20.5	1080	1.94	68.5	1080	3.89
DSP	4	2760	0.25	92	2760	3.33

5. Conclusion

This work presented the design and implementation of the Accumulation and Preprocessing modules forming the front end of the PMI onboard data processing pipeline for ESA's Vigil mission. The FPGA-based IP cores efficiently convert raw filtergrams into calibrated Stokes parameters while meeting all requirements on throughput, latency, and resource use. The results demonstrate that the implemented solution satisfies the performance objectives and is suitable for integration in the flight processing system.

Acknowledgments. This work was carried out in the framework of the International Max Planck Research School (IMPRS) and Technical University, Braunschweig

References

del Toro Iniesta, J. C. 2003, Introduction to Spectropolarimetry (Cambridge: Cambridge University Press)

Muraleedharan, D., et al. 2025, in Proc. European Data Handling & Data Processing Conference (EDHPC). Accepted for publication

Solanki, S. K., et al. 2020, A&A, 642, A11. eprint: 1903.11061

Staub, J., et al. 2020, Journal of Space Weather and Space Climate, 10, 54

Astronomical Data Analysis Software and Systems XXXV
*ASP Conference Series, Vol. **VOL** Compiled: Tuesday 30th June, 2026@ 01:30*
K. Polsterer, D. Bohlender, S. Gwyn, A. Hincks and P. Teuben, eds.
*© **YEAR** Astronomical Society of the Pacific*

ZZZDEBUG!030.tex
Bryant, A.|textbf
Früh, J.
Iserlohe, C.
Knieling, B.

The FIFI-LS Pipeline at the SOFIA Data Center - New Flux Calibration and Improvements

Aaron Bryant, Jonas Früh, Christof Iserlohe, and Bastian Knieling

SOFIA Data Center, Institut für Raumfahrtsysteme, Universität Stuttgart, Germany; bryant@irs.uni-stuttgart.de

Abstract. Among the pipeline enhancements being actively developed at the SOFIA Data Center (SDC) is a new flux calibration for SOFIA's field imaging spectrometer FIFI-LS. Calibration is performed by comparing over 7 years of far-infrared measurements of Mars with known flux models. The methodology improves upon previous calibrations by utilising a fully time-resolved model of the Mars surface brightness, a flux-conserving spatial/spectral resampling algorithm, and photometric techniques that account for the spatially uneven and undersampled Mars disk.

The resulting flux "response curves" show an improved accuracy and reduced variation between SOFIA flights, and better reflect the instrument characterisation than their predecessors. This raises the integrity of the final data products - crucial astrophysical diagnostics such as the fine structure cooling lines [CII] and [OI].

1. Introduction

The FIFI-LS data reduction pipeline, part of the "Redux" Pipeline package, is modularised into a series of stages, one of which calibrates the far-infrared flux density from instrumental to observational units. This is accomplished via division by instrumental response curves with units of ADU/Hz/Jy, derived from observations of "flux standards". At the wavelengths at which FIFI-LS operates, there are very few stars bright enough to yield high signal-to-noise data useful for flux calibration purposes. Therefore, observations of asteroids, planets, and planetary moons are used, along with models of such objects, to derive the response curves. Since the observed fluxes of such solar system objects vary with time, the models must be generated for the time of each specific observation.

To date, observations of Mars have been used as the primary flux calibration source, as shown in Fadda et al. (2023). The methodology described therein presents a number of potential drawbacks, which will be described in the subsequent sections. Given these factors, the SDC resolved to use a new Mars reduction and resampling algorithm (section 2) and time-resolved Mars model (section 3) in order to generate updated instrumental response curves (section 4). This is part of a larger effort to develop and improve the FIFI-LS pipeline in preparation for a full re-reduction of the SOFIA data archive (section 5).

2. Mars Observations

Between 2014 and 2022, FIFI-LS observed Mars over the entire wavelength range of the instrument, covering all combinations of filter and dichoric configuration. Mars observations are reduced using the latest FIFI-LS pipeline at the SDC, which includes updated ATRAN atmospheric correction profiles. Previous reductions used ATRAN files containing identified errors, leading to artefacts in the final response curves. The Mars datasets are grouped by nod cycle, comprising nod-subtracted measurements spanning 3-15 seconds integration time. They are then reduced to wavelength and atmospheric-calibrated 2-dimensional data arrays, representing the spatial-spectral arrangement of the FIFI-LS detectors. In this scheme, spatial pixels are referred to as "spaxels", whereby each spaxel is further divided into a number of spectral pixels, or "spexels".

Unlike previous flux calibration methods, which resample using 3D interpolation, the SDC employs a drizzle-like algorithm to resample the Mars data onto a regular spatial-spectral grid. Testing has proven the method's robustness in conserving fluxes between the input and output grids. However, Mars flux is potentially lost due to non-functional pixels and lack of spatial dithering. To mitigate this, a spaxel-wise linear interpolation is performed over dead spexels, based upon a flight-wise characterisation of the detector arrays (Figure 1).

Following this, the drizzle-like resampling is performed, whereby the input signal from each spaxel-spexel combination is redistributed onto the output grid according to the overlapping area in spatial-spectral space. Fluxes may be extracted from the resampled Mars data using either aperture photometry or a simple summation of all pixels within the output grid. Investigation into the two methods is ongoing.

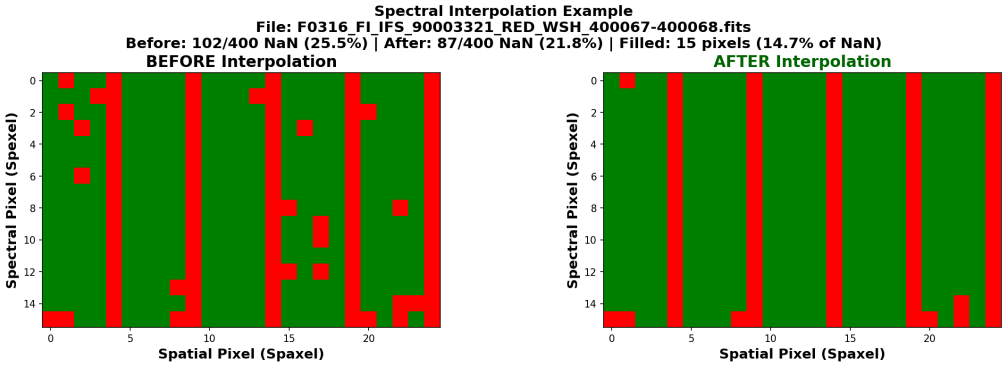


Figure 1. Example of spatial-spectral Mars dataset prior to (left) and after (right) spectral interpolation.

3. Modelling the Mars Brightness

Measured FIFI-LS fluxes are compared with model fluxes derived from the brightness model of E. Lellouch and H. Amri[1]. Previous flux calibration iterations resolved this

[1] https://www.lesia.obspm.fr/perso/emmanuellellouch/mars/.

model to a single date, and subsequently scaled flux curves according to the apparent Mars radius at the time and date of observation. These curves were interpolated onto the wavelength axes of each Mars observation. However, by fully time-resolving the model, second order effects in the Mars model can be accounted for, including surface topology and the Martian phase.

For each spexel row of the resampled data grid, the timestamp and wavelength are used to generate a modelled Mars flux, and consequently the instrumental response. If the above-described resampling algorithm results in a changed wavelength axis, then the model is fully recomputed to the new timestamp-wavelength combinations. Examples of the time-resolved modelled Mars fluxes are shown in Figure 2.

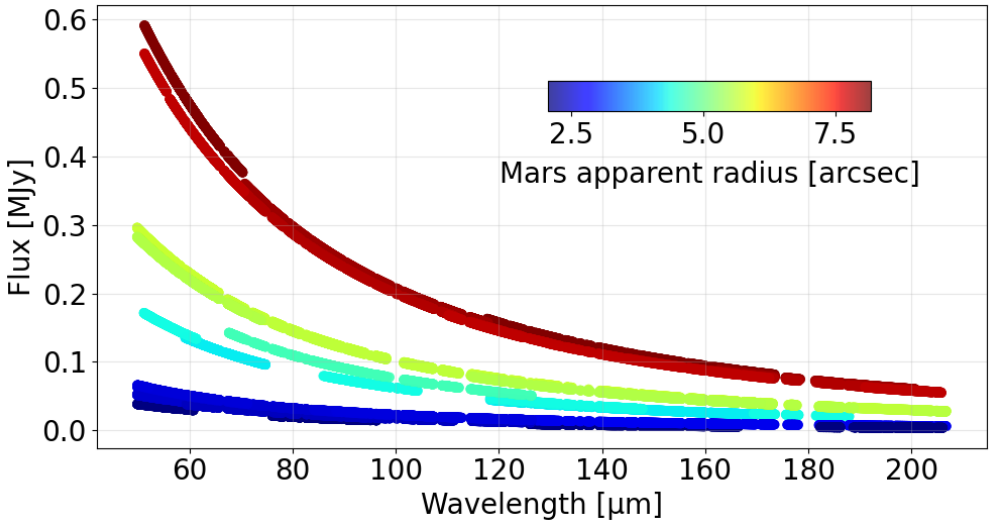


Figure 2. Modelled Mars flux curves at all wavelength-timestamp combinations corresponding to FIFI-LS Mars observations.

4. Generating Response Curves

Following the derivation of observed and modelled Mars fluxes, the instrumental response datapoints are generated via division of the former by the latter. Responses are grouped by instrument configuration and used to generate a smoothed response curve via a robust polynomial fitting algorithm. An example of one response curve is shown in Figure 3. Outlier rejection is performed to remove datapoints affected by residual atmospheric artefacts or other systematic errors. Previous methods applied flight-wise scaling factors to reduce inter-flight scatter. Preliminary results indicate that the new methodology reduces such scatter, and as such, flight-wise scaling may no longer be necessary.

Some scatter is also seen between datapoints within a flight, after outlier detection. These may be caused by edge effects due to the relative offsets between the wavelength axes from one spaxel to another in the input grid. Discarding the outermost spexel rows has proven to be a robust method of mitigating this effect and reducing the intra-flight scatter.

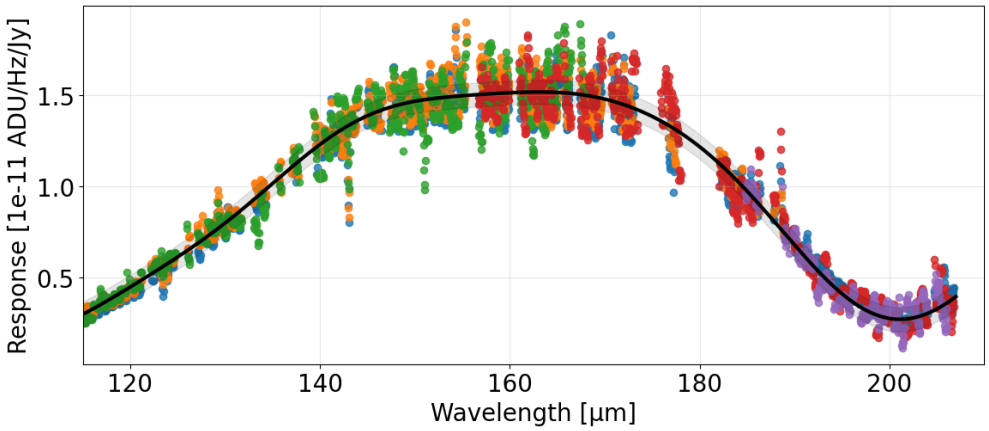


Figure 3. Example instrumental response datapoints and fitted response curve derived from multiple FIFI-LS flights within a common instrument configuration.

5. The FIFI-LS Pipeline

The SDC is implementing several additional improvements to the FIFI-LS pipeline. These include an automated determination of water vapour from time-resolved ECMWF satellite data (Fischer et al. 2021; Iserlohe et al. 2021, 2022), writing of data quality metrics and telescope pointing determination, and generation of science-ready data products, such as spectral line fits and map mosaics. Documentation and support for non-standard observing modes are also being developed, some of which have already demonstrated significant improvements in science data quality (Fischer et al. 2025). Redux and the FIFI-LS pipeline are publicly available on GitHub[2].

Acknowledgments. The authors thank Dr. Christian Fischer and Dr. Emmanuel Lellouch for their active assistance in the flux calibration efforts. The SDC is funded by the DLR Space Administration under grant number FKZ 50OK2404.

References

Fadda, D., Colditz, S., Fischer, C., Vacca, W. D., Chu, J., Clarke, M., Klein, R., Krabbe, A., Minchin, R., & Poglitsch, A. 2023, The Astronomical Journal, 166, 237

Fischer, C., Iserlohe, C., Vacca, W., Fadda, D., Colditz, S., Fischer, N., & Krabbe, A. 2021, Publications of the Astronomical Society of the Pacific, 133, 055001

Fischer, C., Madden, S. C., Krabbe, A., Polles, F. L., Fadda, D., Tarantino, E., Galliano, F., Chen, C.-H. R., Abel, N., Beck, Á., Belloir, L., Bigiel, F., Bolatto, A., Chevance, M., Colditz, S., Fischer, N., Green, A., Hughes, A., Indebetouw, R., Iserlohe, C., Kaźmierczak-Barthel, M., Klein, R., Lambert-Huyghe, A., Lebouteiller, V., Mikheeva, E., Poglitsch, A., Ramambason, L., Reach, W., Rubio, M., Vacca, W., Wong, T., & Zinnecker, H. 2025, Astron. Astrophys., 702, A273

Iserlohe, C., Fischer, C., Vacca, W. D., Fischer, N., Colditz, S., & Krabbe, A. 2021, Publications of the Astronomical Society of the Pacific, 133, 055002

Iserlohe, C., Vacca, W. D., Fischer, C., Fischer, N., & Krabbe, A. 2022, Publications of the Astronomical Society of the Pacific, 134, 085001

[2] https://github.com/SOFIA-Data-Center/sofia_redux

Astronomical Data Analysis Software and Systems XXXV
*ASP Conference Series, Vol. **VOL** Compiled: Tuesday 30th June, 2026@ 01:30*
K. Polsterer, D. Bohlender, S. Gwyn, A. Hincks and P. Teuben, eds.
*© **YEAR** Astronomical Society of the Pacific*

ZZZDEBUG!039.tex
Fernique, P.|textbf
Allen, A.
Baumann, M.
Boch, T.
Bot, C.
Pineau, F.
data!Hierarchical Progressive Surveys (HiPS)
International Virtual Observatory Alliance (IVOA)
PORTAL! ascl:2003.006
HEALPix! ascl:1107.018
CUBE! ascl:1805.018
Aladin! ascl:1112.019
CARTA! ascl:2103.031
DS9! ascl:0003.002

Big data exploration with HiPS3D - a hierarchical visualisation solution for cubic surveys

Pierre Fernique, Mark Allen, Matthieu Baumann, Thomas Boch, François Bonnarel, Caroline Bot, François-Xavier Pineau

Université de Strasbourg, CNRS, Observatoire astronomique de Strasbourg, UMR 7550, 11 rue de l'Université, F-67000 Strasbourg, France

Abstract. With the new generations of large-scale surveys, we are faced with an avalanche of data that are no longer "images" but "cubes", and whose third dimension is either temporal or spectral. In this new area, traditional hierarchical science platform visualisation methods must evolve to exploit this third dimension. Building on the Hierarchical Progressive Survey method, we present new results from a study and implementation of extending this visualisation method to a third physical dimension: the HiPS3D solution. We will explain the principles behind this generalization, the scientific and technical choices, and we will present the new prototype tools: Hipsgen for the generation and Aladin (Desktop & Lite) for the visualisation and integration in various science platform solutions. We will report on the first statistics obtained on test cube surveys. Finally, we will give an overview of the prospects for a large-scale implementation of this method for existing cubic surveys such as ALMA, or future ones such as SKA.

1. The big data vizualisation challenges

Logically, visualising astronomical data requires access to the pixels produced by observational instruments. Therefore, the network — particularly the internet — is the determining factor in transmitting pixels from the "telescope" to the "astronomer", from the "server" to the "client". If the network capacity is insufficient for the volume of pixels involved, the astronomer will be forced to perform "remote visualisations". Due to the explosion in data volumes of recent surveys, traditional application clients such as DS9 are not adapted, and new modes of access via remote sessions are emerging, such as remote SSH sessions, Jupiter notebooks and remote Carta sessions. However, these solutions have two major limitations. Firstly, network latency introduces a delay for each action (movement, zooming, changing the colour map, etc.), which makes such visualisation tools slow. Furthermore, the resources required for these "remote sessions" (RAM, CPU and GPU) must be scaled to support all simultaneous "remote clients". The resulting cost will necessarily limit scalability considerably.

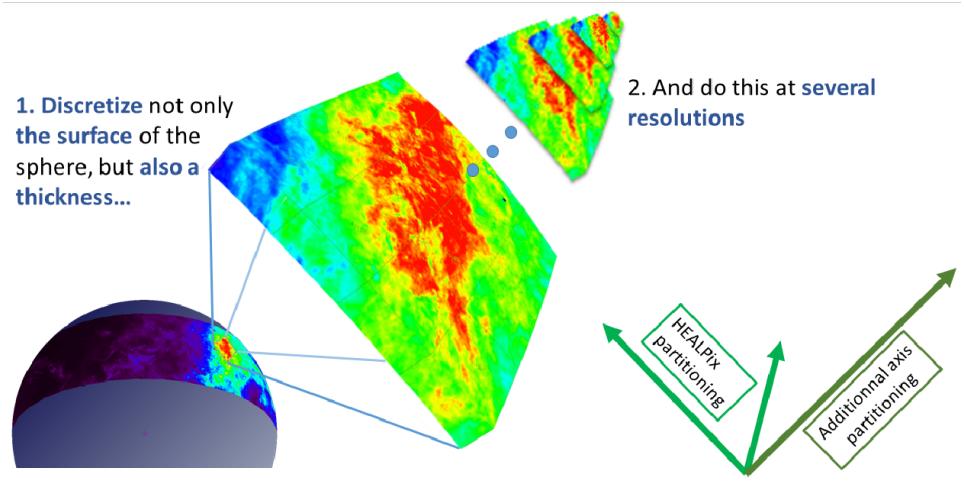
However, a third approach is possible. This is based on the simple principle that only the pixels being viewed need to be transmitted, rather than all the observations. This principle has been implemented in the HiPS (Hierarchical Progressive Surveys) method, which was invented by the Centre de Données Astronomiques de Strasbourg and endorsed/standardised by the IVOA in 2017 (Fernique et al. 2017). To enable fast and efficient access to the displayed pixels, the original observations are organised into

network-size-compatible "tiles". This step is performed in advance using a global mosaicking/resampling based on a HEALPix tessellation of the celestial sphere at different resolutions. Thanks to that we have a global addressing method based on two numbers - the HEALPix order of the resolution and the HEALPix index of the tile - allowing any clients to directly access the tiles covering the current field of view. The HiPS method has been implemented in a wide range of professional and amateur visualisation tools, including: Aladin, ESAsky, Firefox, RSA Cosmos, Digistar, Stellarium and WWT. By 2025, over 25 servers worldwide were responding to millions of requests per day for HiPS tiles from 1,400 surveys including DSS, 2MASS, GALEX, SDSS, Pan-STARRS.

Over the past 10 years, HiPS has demonstrated its full potential, but it is now facing the recent advent of sky surveys that produce cubic data, where the third dimension can be time (e.g. Rubin) or frequency (e.g. MUSE, MEERKAT). As part of its involvement in the SKA SRCnet project, this article outlines the proposals and tests conducted by the CDS to expand the original HiPS method to cubic surveys.

2. The HiPS 3D principle

The main HiPS3D idea is to apply discretisation simultaneously to the original cubic data in both space and the additional dimension (time or frequency). In order to achieve this, in addition to using HEALPix for the spatial data, a hierarchical partitioning function had to be defined for the additional axis. For this purpose, we propose reusing the functions already presented and partially standardised by the IVOA (Fernique et al. 2022) for describing spatiotemporal and spatiofrequential coverage.



This time, the objective is to achieve direct addressing of cubic tiles using a quadruplet comprising the order and spatial index (HEALPix), as well as the order and index in the third dimension (e.g. OrderS/NpixS_OrderF/NpixF). As with traditional HiPS, the resulting cubic tiles are stored as files in a conventional file system, organised into directories. These tiles may have been generated using cubic FITS encoding to preserve all of the original pixel dynamics. Alternatively, they may have been generated in compressed mode (PNG, JPEG or WebP checkerboard) for use with basic exploration tools.

3. Prototypes tools and test surveys

To evaluate the relevance of the proposed HiPS 3D solution, the CDS's Hipsgen tool (Bot et al. 2022) for generating HiPS was expanded in 2025, as was the Aladin Desktop visualisation client. This prototyping phase focused on cubic spatio-frequency surveys. Tests were carried out on cubes surveys of different types: GALFAHI spatial mosaic covering a third of the sky; MUSE pointed observations excerpt at various small sky locations; ASKAP frequency band combination at the same sky location; MEERKAT pointed observation at various "space & frequency" location; ALMA excerpt on spatial and frequency combination for all M83 observations; SKADC2 huge original cube from SKA simulation; and more: LGBSHI, DHIGLS, MANGA excerpt, HARMONI...

The table below shows the times and volumes required for the GALFAHI and ALMA tests. We have deliberately chosen an output resolution that is as close as possible to that of the original data. The figures shown are for the FITS tile generation process, which preserves all the dynamics of the original pixels, as well as the next step of adding the compressed tiles.

Input data	**HiPS tiles**	**Time**	**Nb. files**	**Final volume**
GALFAHI	FITS	32 min 05 s	140 502	250 GB
(225 cubes, 225 GB)	PNG	+2 min 48 s	140 502	7.8 GB
	JPEG	+1 min 25 s	140 502	4.2 GB
ALMA	FITS	53 min	10 060	28.3 GB
(88 cubes, 675 GB)	PNG	+1 min 21 s	10 060	3.3 GB

In fact, the final volume of the HiPS3D obtained, and therefore its processing time, is determined by four factors: the choice of the final spatial and frequency resolutions; whether or not to compress the tiles; the portion of the original data overlays; the HiPS3D "pyramid" of pre-calculated tiles for lower resolutions. In the tests carried out, this last factor actually only accounted for 15% of the final volume. It turns out that, for visual exploration, simultaneously degrading the spatial and frequency resolutions is sufficient to navigate efficiently in the HiPS3D data.

4. HiPS3D tools

As of the end of November 2025, five HiPS3D-compatible tools were available for testing, from CDS and other contributors: Hipsgen: HiPS generation (public version); Aladin Desktop: standalone HiPS visualization (version beta 12.6); Aladin Lite: Web & jupyter HiPS visualisation (prototype v3.7); Astropy/reproject/Glue: Generation & visualisation (Robitaille et al. 2019); Hips2fits3D: server side extraction of any cube at any resolution from a HiPS3D (Boch et al.).

Various alternatives are conceivable for vizualizing HiPS3D tiles. In the case of Aladin Desktop and Aladin Lite, the choice was made to simultaneously display the spatial view centered on a celestial position, associated with a given frequency channel, and superimposed on it, the spectral view corresponding to the position of the central pixel of the spatial view. Sliding the spatial view updates the spectral view, and conversely, sliding the spectrum updates the channel of the spatial view. Zooming operates on both views simultaneously.

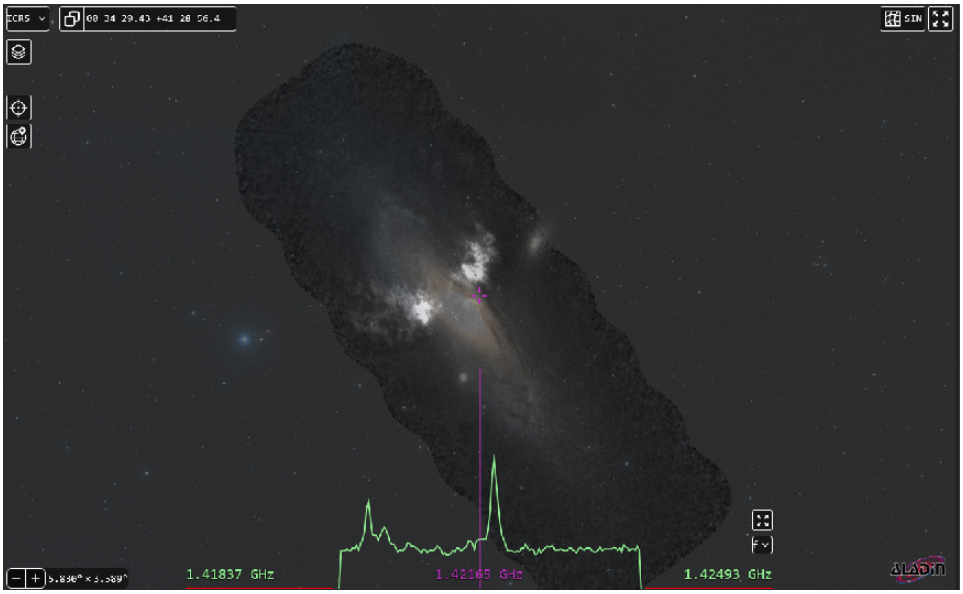


Figure 1. HiPS3D Local Group L-Band HI Survey overlayed with HiPS DSS color in Aladin Lite (Boch & Fernique 2014)

5. Conclusion

In conclusion, let us consider the case of an exceptionally large spectral-imaging cube survey, on the order of several hundred petabytes, as anticipated for SKA-scale observations. By decreasing the spatial resolution by two HiPS orders and retaining only the compressed tiles, the additional storage requirement is limited to approximately 0.2%. With only a few additional terabytes of storage, this HiPS3D method enables the visualisation of petabyte-scale cube surveys from any access portal, including mobile platforms.

References

Boch, T., Baumann, M., & Fernique, P.in ADASS XXXV, edited by K. Polsterer, & S. Wagner (San Francisco: ASP), vol. TBD of ASP Conf. Ser., 999 TBD

Boch, T., & Fernique, P. 2014, in Astronomical Data Analysis Software and Systems XXIII, edited by N. Manset, & P. Forshay, vol. 485 of Astronomical Society of the Pacific Conference Series, 277

Bot, C., Fernique, P., Oberto, A., Durand, D., Boch, T., Allen, M., Buga, M., & Bonnarel, F. 2022, in Astronomical Data Analysis Software and Systems XXX, edited by J. E. Ruiz, F. Pierfedereci, & P. Teuben, vol. 532 of Astronomical Society of the Pacific Conference Series, 467

Fernique, P., Allen, M., Boch, T., Donaldson, T., Durand, D., Ebisawa, K., Michel, L., Salgado, J., & Stoehr, F. 2017, HiPS - Hierarchical Progressive Survey Version 1.0, Tech. rep.

Fernique, P., Nebot, A., Durand, D., Baumann, M., Boch, T., Greco, G., Donaldson, T., Pineau, F.-X., Taylor, M., O'Mullane, W., Reinecke, M., & Derrière, S. 2022, MOC: Multi-Order Coverage map Version 2.0, IVOA Recommendation 27 July 2022

Robitaille, T., Beaumont, C., Qian, P., Borkin, M., & Goodman, A. 2019, glueviz v0.15.2: multidimensional data exploration

Astronomical Data Analysis Software and Systems XXXV
*ASP Conference Series, Vol. **VOL** Compiled: Tuesday 30th June, 2026@ 01:30*
K. Polsterer, D. Bohlender, S. Gwyn, A. Hincks and P. Teuben, eds.
*© **YEAR** Astronomical Society of the Pacific*

ZZZDEBUG!040.tex
Nieto, S.|textbf
de Teodoro, P.
Oliveira, J.
Fernandez, J.
Rios, C.
Prados, J.
Olmedo, V.
Canovas, J. L.
Saiz, J.
Fernandez, V.
Prieto, R.
Ruiz, E.
Perez, P. A.

Euclid Science Archive technical evolution to the Petabyte scale

Sara Nieto,[1] Pilar de Teodoro,[2] Joaquim Oliveira,[2] Jorge Fernandez,[1] Carlos Rios[1], Jordi Prados,[1] Victor Olmedo,[1] Jose Luis Canovas,[1] Jaime Saiz,[1] Victor Fernandez,[1] Roberto Prieto,[1] Enrique Ruiz,[1] and Pablo A. Perez[3]

[1]*Starion for European Space Agency (ESA), ESAC; Villanueva de la Cañada, Madrid, Spain;*

[2]*Aurora Technology for European Space Agency (ESA), ESAC; Villanueva de la Cañada, Madrid, Spain;*

[3]*Starion for European Space Agency (ESA), ESAC; Villanueva de la Cañada, Madrid, Spain;* `Pablo.perez@ext.esa.int`

Abstract. ESA's Euclid cosmology mission was launched in 2023 and during its five year life time it will observe one third of the sky with unprecedented resolution with its optical and near infrared instruments.

The Euclid data will be hosted by the Euclid Science Archive, within the ESAC Science Data Centre (ESDC) in Madrid. The ESDC is in charge of the development, operations and maintenance of the science archives for all the ESA astronomy, planetary and heliophysics missions, as well as for the ESA Human Robotic and Exploration investigations.

The Euclid Science Archive released the Quick Release 1 (QR1) in May 2025, making available to the public 35 Terabytes of data. The next major milestone is the Internal Data Release 1 (IDR1) in October 2025, whose data will grow to **2.2** Petabytes. By the end of the mission, over **30** Petabytes of official products from the Euclid mission will be hosted at the Euclid Science Archive.

The purpose of this presentation is to show in detail the software, hardware and architectural progress needed for the critical transition to the IDR1 milestone, and how the Euclid Archive is getting ready to prepare for future, bigger releases.

The topics to be discussed will range from database technology specific for big data, parallelization of download and ingestion of products from several Euclid Data Centres across Europe and the US, as well as custom configuration of filesystem architecture to guarantee a transfer and ingestion performance of up to 70 Terabytes per day.

1. Introduction

The Euclid mission of the European Space Agency (ESA) launched in July 2023. It will investigate the origin of the Universe and the nature of the dark energy, dark matter and gravity. It is expected to be operating for 6 years. It is also expected to generate over 30 PB of processed data by the end of the mission. Therefore, it is a Big Data mission.

All of its data will be made public through the Euclid Data Space [1], that hosts the Euclid Science Archive [2]. The Euclid Science Archive is an interface (both programmatic and interactive) that provides all scientific the data to the users as well as extra functionalities to help the user navigate, operate and interact with the data.

This paper will go into detail about the technical evolutions done in the Euclid Science Archive to improve the Data Transfer throughput to match the requirements for the Euclid Mission.

2. IDR1 Data Release

The **Euclid IDR1** is the first major release of the Euclid mission. It is targeted to Euclid Consortium members and will be made available to the public later in Q4 of 2026 as the **DR1**.

To have a sense of scale of the IDR1, the previous public release for Euclid was the **Q1**, released in March 19th of 2025, with a total size of **35TB**.

All the data of all previous archives in the ESDC[3], 30 years worth of missions data, are around **900TB**.

The IDR1 consists of about **1.5 PB** of data. It is a **major** increase in size for ESDC missions, and so, a significant challenge to be able to process all the release within the schedule of the mission. With the performance previous to this upgrade, it would take **250 days** to complete the ingestion of the data release.

3. EAS[4] Data Transfer Pipeline

The EAS Data Transfer Pipeline consists of four key components: the **MTS** (*Metadata Transfer Service*), the **Ingestor**, a NetApp architecture **filesystem**, and a distributed **Greenplum database**.

3.1. MTS

The MTS is a Java based service in charge of querying the DPS and DSS[5] for relevant products to download, then downloads the XML of the product, to finally download all the associated files for each product in the *staging folder*.

3.2. Ingestor

The Ingestor, in Java as well, is the service in charge of the XML and product processing. It verifies that the XML files in the *staging folder* are compliant with the Euclid Data Model and checks that all their associated files have been downloaded as well.

---

[1] `https://euclid.dataspace.esa.int/data-discovery`

[2] `https://eas.esac.esa.int/sas/`

[3] ESA Science Data Centre

[4] Euclid Archive System

[5] *Data Processing System* and *Distributed Storage System*, part of the Network of Euclid Science Data Center where the Euclid Consortium stores the process data.

Then, it decompresses the (heavy) gzip files, moves all files from the *staging folder* to the *repository volume* and inserts the metadata in the database.

4. Improving the Data Transfer Pipeline

4.1. Improving the infrastructure

There were infrastructure improvements for all ESDC with the intention to improve the performance and throughput of all mission's archives. One was the upgrade of the Network Interface from 10 Gbps to **100 Gbps**. The other one was contacting with **NetApp** to optimize the NetApp configuration for ESDC needs.

4.2. Improving the code

The other major challenge was to improve the code of the EAS, in particular of its two main services, the MTS and Ingestor. The requirement was moving from a non parallelized code, to a fully parallelized and multi-threaded code. The team had a clear direction to move to, and that was to take advantage of Apache Camel[6], its parallelization tools and efficient message routing management.

Coming from a working non parallelized code to a parallelized one implies a major refactor and a big overhaul of the whole services. This is a big risk, as it could take too long to have in time for the mission requirements; also, the ingestion needed to keep going while the team worked on the upgrade.

We made the decision to take an **incremental approach** with this upgrade. We would divide the improvements in several software releases.

4.2.1. Incremental approach - MTS

Between the Ingestor and the MTS, the latter is the easier use case as its main function is to manage the download of files. To do the upgrade, the team did a whole refactor of the code. We moved away from big Java functions to small, specific functions to match the best practices of Apache Camel. All the steps of the process were moved to Camel routes and Camel processes that called small and lean Java functions. Taking advantage of a few Camel components (like *toD* to direct the process flow and *seda* to have multi-threaded runtime, the MTS improved up to **4 times** its download throughput on a new release: from an average of 1 Gbps to **4 Gbps**.

4.2.2. Incremental approach - Ingestor

After the release of the MTS improvements, the Ingestor was next. It is the most complex piece of software of the EAS. It has two main functions: first, processes the XML files in the *staging area* and prepares the metadata to insert in the database; second, moves the correctly processed files from the *staging area* to the *repository volume*.

The XML processing is the most complex function of the two, as it has a different use case for each of the tenths of different products the EAS ingests. Therefore, the team decided to take on the **file movement first**. This function is the same for each product type, and is also very similar to the MTS use case: in the end, it involves

[6] Apache Camel documentation: `https://camel.apache.org/docs/`.

moving files from an origin to a destination. The team knew how to parallelize this use case, and it will improve the performance for the ingestion of all products.

Lastly, the parallelization of the XML processing and database insertion. This was a complex process with a lot of nuances and race conditions. The team further divided the upgrade, focusing first on the most critical products, the ones that were causing the biggest bottleneck. This allowed the team to attack a very complex process from simpler, isolated points.

4.3. Final picture

The outcome of this improvement is that, after several operational releases, all EAS data products benefit from being parallelized and multi-threaded. A nice benefit of the incremental approach is that the team developed the parallelization to be configurable via properties stored in the database, so once the upgrade has been completed the number of threads is **dynamically configurable** per product type. This allows for a lot of fine tuning for throughput optimization.

The final throughput numbers after the upgrade are the following: from an average of 1 Gbps, to 4 Gbps after the MTS improvement, to a final high performance of **10 Gbps**, with a peak of 11.8 Gbps. This results in about **64TB ingested per day**, more than 10 times the original performance. With this performance, the data release was ingested in **two months**.

5. Next challenges

This upgrade provides us with a strong foundation to face the next challenges of the Euclid mission. After the **DR1** with **2.1 PB** of data will come the **DR2**, scheduled for 2029, with **6.3 PB**. Then, the last major release will be the **DR3**, expected for 2031, with **12.6 PB**. Thanks to the tools developed in this upgrade, and the knowledge acquired by the team, we are confident the code can be further optimized to achieve these major milestones in time.

Acknowledgments. I would like to thank the Euclid Consortium, the Euclid mission management, all the colleagues from ESDC and the SCO08 service who helped on this major milestone. Lastly, I would like to thank the dedicated members of the Astro Survey Subarea who tirelessly worked to get this upgrade in time and in excellent form.

Astronomical Data Analysis Software and Systems XXXV
*ASP Conference Series, Vol. **VOL** Compiled: Tuesday 30th June, 2026@ 01:30*
K. Polsterer, D. Bohlender, S. Gwyn, A. Hincks and P. Teuben, eds.
*© **YEAR** Astronomical Society of the Pacific*

ZZZDEBUG!041.tex
Raison, F.|textbf
Bassler, T.
Meneses, N. C.
Friedl, P.
Gidofalvy, A.
Schwinger, M.

prosEO: A Cloud-Native Processing Framework for Hybrid Cloud–HPC Astronomy Pipelines

Frédéric Raison[1], Thomas Bassler[2], Nicolás Corti Meneses[1], Peter Friedl[1], Anett Gidofalvy[1],and Maximilian Schwinger[1]

[1]*Deutsches Zentrum für Luft- und Raumfahrt (DLR), German Aerospace Center,*
Earth Observation Center, German Remote Sensing Data Center, International Ground Segment (IBS)
[2]*Dr. Bassler & Co. Managementberatung GmbH*

Abstract. Upcoming astronomy missions face rapidly increasing data volumes, complex processing chains, and the need to integrate heterogeneous computing infrastructures while maintaining operational sustainability. Similar challenges have already been addressed in Earth Observation (EO) missions through cloud-native processing frameworks. We present prosEO (Processing System for Earth Observation), a mission-agnostic, operational processing control system developed at DLR and currently used for multiple Sentinel missions. prosEO provides metadata-driven workflow orchestration, elastic scalability, and multi-infrastructure deployment capabilities. In this contribution, we propose prosEO as a candidate framework for technology transfer to astronomy data processing, enabling hybrid cloud–HPC architectures, near-real-time event processing, and large-scale reprocessing campaigns. We describe the system architecture, highlight operational performance from EO missions, and discuss how prosEO can be adapted to future astronomy missions.

6. Introduction

Astronomy data processing is entering an era of unprecedented scale and complexity. Survey missions and observatories generate petabyte-scale datasets, require both near-real-time processing for alerts and long-term reprocessing campaigns, and increasingly rely on distributed computing infrastructures. Traditional mission-specific, monolithic processing pipelines are becoming costly to develop, operate, and evolve.

Earth Observation missions have faced similar challenges over the last decade. In response, DLR developed prosEO, a cloud-native processing control system designed to orchestrate large-scale EO data processing across heterogeneous infrastructures. prosEO is currently operational for Sentinel missions and supports near-real-time processing, systematic production, and bulk reprocessing.

This paper explores how prosEO can be transferred from EO to astronomy, offering a reusable, scalable, and sustainable processing framework.

7. prosEO System Overview

prosEO is a mission-agnostic, service-oriented processing framework designed to manage end-to-end data production, from raw ingestion to user-level products. The system is based on a microservices architecture and is fully cloud-native.

Key features include:

- **Metadata-driven workflow orchestration:** Processing workflows are dynamically generated based on requested output products and their dependencies.
- **Automatic dependency resolution:** prosEO constructs processing graphs backwards from target products, ensuring only required steps are executed.
- **Container-based execution:** Processing algorithms are executed as isolated container jobs, enabling portability and reproducibility.
- **Infrastructure agnosticism:** prosEO can be deployed on public clouds, private clouds, and on-premises infrastructures.

The system core consists of a production request system, product metadata management, a processing execution engine, and orchestration services.

8. Operational Experience in Earth Observation

prosEO is operational in several EO mission contexts, including Sentinel-5P, Sentinel-1, and Sentinel-4 commissioning activities, which are part of the European Union's Copernicus Earth Observation Programme (cop 2024) implemented by ESA and EUMETSAT. A notable example is the reprocessing of approximately 1.6 PB of Sentinel-5P data over 4.5 years of mission operations. This reprocessing campaign was completed in six months using four virtual data centers with 125 worker nodes each.

Operational experience demonstrates:

- Virtually unlimited horizontal scalability.
- Robust handling of systematic production and on-demand processing.
- Improved traceability and provenance through metadata tracking.
- Modular mission-specific extensions without impact on the prosEO core.

These characteristics closely align with the needs of future astronomy missions.

9. Proposed Architecture for Astronomy Missions

For astronomy applications, prosEO can act as a central orchestration layer coordinating both cloud-based processing facilities and high-performance computing (HPC) systems. Figure 1 illustrates a conceptual architecture.

Instrument data from space- or ground-based observatories are ingested into a data repository. prosEO monitors data availability and triggers workflows for either near-real-time processing (e.g., alerts, quick-look products) on cloud processing facilities or heavy computational tasks on HPC systems.

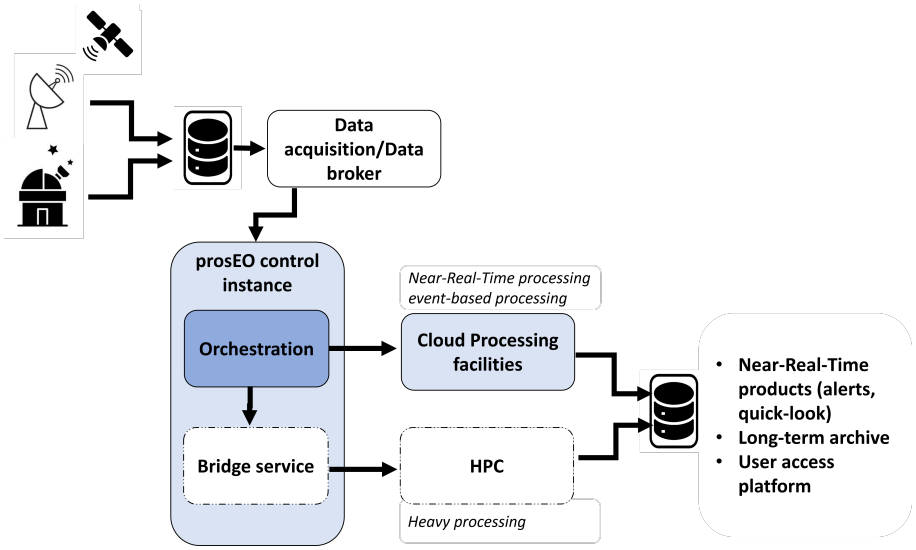


Figure 1. Conceptual prosEO-based architecture for astronomy missions, integrating cloud processing facilities and HPC resources.

A dedicated HPC bridge service translates prosEO job requests into batch scheduler submissions (e.g., Slurm), handling authentication, resource allocation, and status feedback. Output products are stored in long-term archives and disseminated through user access platforms.

10. Benefits for Astronomy Data Processing

Adopting prosEO for astronomy offers several advantages:

- **Reduced development effort:** Missions can focus on science algorithms while reusing an existing, operational framework.
- **Hybrid infrastructure support:** Seamless integration of cloud elasticity with HPC performance.
- **Scalability and sustainability:** Efficient resource usage and reduced duplication across missions.
- **Event-driven processing:** Natural support for alerts and time-critical workflows.

The proposed approach aligns with emerging trends in astronomy data systems, including federated processing, FAIR data principles, and sustainable computing.

11. Conclusion and Outlook

prosEO represents a mature, operational solution for large-scale data processing, proven in EO missions and well aligned with the evolving needs of astronomy. We propose

prosEO as a candidate framework for technology transfer, enabling hybrid cloud–HPC architectures for future astronomy missions.

Ongoing work focuses on adapting prosEO to HPC environments, developing standardized bridge services, and engaging with the astronomy community to validate use cases. We believe that cross-domain reuse of processing frameworks such as prosEO can significantly reduce cost and risk while accelerating scientific exploitation.

Acknowledgments. The authors acknowledge the support of DLR and its partners in the development and operation of prosEO.

References

2024, The copernicus programme, `https://www.copernicus.eu`

Astronomical Data Analysis Software and Systems XXXV
*ASP Conference Series, Vol. **VOL** Compiled: Tuesday 30th June, 2026@ 01:30*
K. Polsterer, D. Bohlender, S. Gwyn, A. Hincks and P. Teuben, eds.
*© **YEAR** Astronomical Society of the Pacific*

ZZZDEBUG!043.tex
Emonts, B. H. C.|textbf
FOOBAR!conference!ADASS 2019
FOOBAR!organisations!ASP
FOOBAR! ascl:1101.010

Role of General Users in the Lifecycle of Scientific Software

Bjorn Emonts,[1,2] and the CASA Team

[1]*National Radio Astronomy Observatory, 520 Edgemont Road, Charlottesville, VA 22903, USA;* bemonts@nrao.edu

[2]*CASA User Liaison;* casa-feedback@nrao.edu

Abstract. In science, the lifecycle of software products is typically managed with limited resources while facing unlimited demand. Scientific software requirements are necessarily often dominated by internal project specifications and deadlines, but these internal priorities, while beneficial for the community as a whole, do not always align with the individual needs of our ultimate customers: general users. For software products to have the broadest reach, ideally the general user community should be involved in all aspects of the data lifecycle, but reality is that user expectations need to be managed. Based on the lifecycle of the Common Astronomy Software Applications for radio astronomy (CASA), we will show avenues for software teams to interact with general users, even when facing limited resources for user support. We will discuss how involvement of users and user groups in prioritizing software development can benefit both the user community and the software teams. The contents of these proceedings were presented at the 35th conference on Astronomical Data Analysis Software & Systems (ADASS XXXV).

1. Introduction

Astronomy is a field of research with a large scope and reach (Astro2020: National Academies of Sciences, Engineering, and Medicine 2023). It relies on state-of-the-art observatories and world-class computing facilities, as well as the astronomical software packages that are needed to process and analyze data from these facilities. An historic overview of the widespread introduction of software in astronomy was published by Shortridge (2001). Astronomical software is often highly specialistic with a limited user-base, consisting of mostly scientists and software engineers. This puts limits on the resources that are available for the development, maintenance, and support of astronomical software.

The Common Astronomy Software Applications (CASA) is a versatile software package for the processing of radio astronomical data from various radio observatories (CASA Team et al. 2022). It has a wide user-base among the community, and is managed by an international development team. In these proceedings, we describe the role that general users play in the lifecycle of the CASA software. Section 2 gives a high-level overview of CASA and its development lifecycle. Section 3 highlights the role that general users play in the development lifecycle, and how stakeholder representation provides critical input for the long-lived health and versatility of the software. In Sect. 4, we address how the CASA team balances user support with available resources.

Section 4 concludes with a brief outlook on the future of CASA, while Sect. 6 provides an overview of CASA resources that are available to the user community.

2. The CASA Software for Radio Astronomy

CASA is the *Common Astronomy Software Applications*, which is an open-source software package for radio astronomy (CASA Team et al. 2022). CASA is the official software for the Karl G. Jansky Very Large Array (VLA), Atacama Large Millimeter/submillimeter Array (ALMA), and Nobeyama 45m Telescope, but is often used also for the calibration and imaging of data from other radio telescopes. CASA can process interferometric, single-dish, and Very Long Baseline Interferometry (VLBI) data.

CASA can be run interactively and through scripting, and a core functionality is to support calibration and imaging pipelines for ALMA, VLA, and Nobeyama (Hunter et al. 2023). CASA is implemented in C++ and accessible through IPython (Perez & Granger 2007). It is built on top of Casacore, which contains the core libraries, and which is a nearly static platform that is used by many radio observatories (Casacore Team 2019). The CASA software consists of the following core functionality:

- *Tools:* basic C++ functions as Python-class objects with *methods* that perform basic operations on the data;
- *Tasks:* user-friendly bundles of tools and Python functionality that perform specific steps in the data processing. CASA tasks contain *parameters* that give users flexibility to control the calibration and imaging;
- *GUIs:* Graphical User Interfaces (GUIs) to visualize and examine data and images;
- *Data Repository:* external data that changes over time, and which is needed for accurate calibration and imaging. This includes reference frames, Earth Orientation parameters, and telescope-specific models.

Historically, CASA has been offered as an all-inclusive package, but since CASA 6 the software has been available in various formats:

- *Monolithic tar-file:* this is the all-inclusive, 'plug-and-play' format that contains a Python interpreter and all necessary libraries;
- *Modular pip-wheels:* this format allows users to run CASA tasks, tools, and GUIs in their own Python environment, or in Jupyter Notebooks or Google Colab;
- *Pipeline version:* certain tar-file versions of CASA are packaged together with the calibration and imaging pipelines for ALMA and the VLA.

CASA is being developed by an international team of around 30 scientists and software engineers based at the National Radio Astronomy Observatory (NRAO), the European Southern Observatory (ESO), the National Astronomical Observatory of Japan (NAOJ), and the Joint Institute for Very Long Baseline Interferometry European Research Consortium (JIVE), under the guidance of NRAO.

2.1. Lifecycle of CASA

The CASA team strives to maintain a "rolling" release schedule, with a new CASA release roughly every three months (Fig. 1). Such a rolling release schedule simplifies and standardizes the development cycle and release process. This quarterly release is available both in monolithic and modular format (see Sect. 2). Once per year, the interferometric pipeline that supports ALMA, VLA, and the VLA Sky Survey (VLASS) is bundled with CASA, and released in monolithic tar-file format. While ALMA, the VLA, and VLASS strive to use the same pipeline, there can be small differences that may lead to different flavors of the pipeline, each with its own CASA release version.

Development priorities are set by CASA management following input from a large number of stakeholders, who provide requirements and feature requests twice a year. This is complemented with a yearly report from the DMS Panel of the Users Committee, which is an advisory committee on the development of software at NRAO, which operates on behalf of the general user community (see Sect. 3.1).

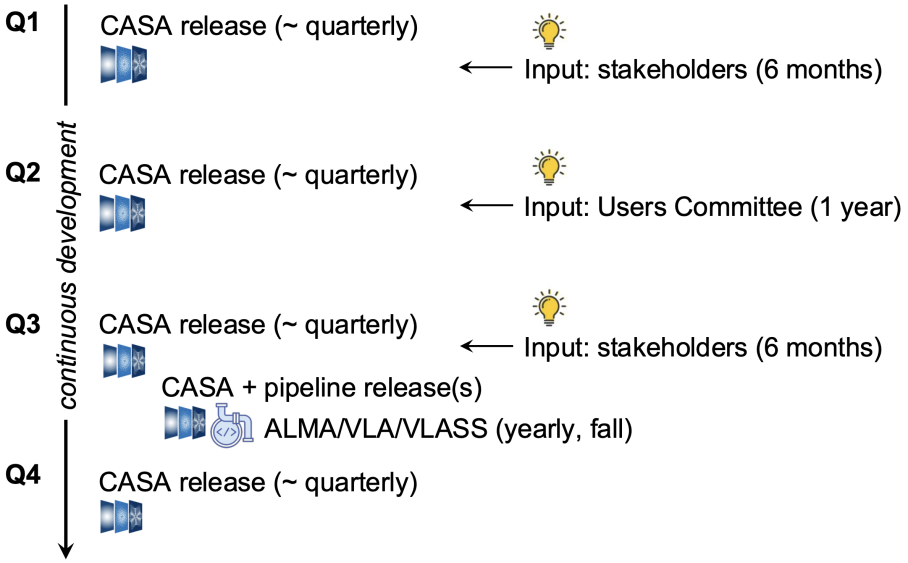


Figure 1. Lifecycle of the CASA software. CASA is released quarterly, with once a year a CASA release bundled with the pipeline for ALMA, VLA, and VLASS. Stakeholder input is gathered every six months, while once a year the DMS Panel of the Users Committee (DPUC) provides a report with recommendations on behalf of the user community.

3. Role of Users in Software Development

General users are the ultimate customers of software products. As such, they provide unique and important input critical for the long-terms health and versatility of the software. This is particularly true if the software is intended for broad use, as is the case for CASA. The problem is that general users form a silent majority that is not as vocal as institutional stakeholders and experts, or power-users. It is good to keep in mind that general users often see scientific software as a means-to-an-end, not the end goal.

A solution to this problem is to capture feedback from the general user community. This can be done passively, for example by collecting software statistics, telemetry data, or crash reports, or actively, through user surveys, review panels, or direct contact (e.g., helpdesk or email). In the case of CASA, we have given the user community a voice as stakeholder, which has resulted in unique insights into essential requirements that might otherwise not have been obtained.

3.1. Users as CASA Stakeholders

As shown in Fig. 2, CASA has a large number of stakeholders. Stakeholders whom we identify in these proceedings as "internal" are ALMA, VLA, VLBA, and groups at NRAO's department of Data Management and Software (DMS). These internal stakeholders are represented by local colleagues, and therefore have a strong voice to advocate institutional requirements. These requirements are often short-term priorities essential for telescope operations, such as specific tasks to support new observing modes, new parameters or defaults for pipeline operations, or bug that block data processing.

It takes a more active approach to obtain feedback from the general users. For CASA, the user community also has a voice as stakeholder, which we here identify as "external" stakeholder, with representation through both the CASA User Liaison and the DMS Panel of the Users Committee (DPUC). The CASA User Liaison (author of these proceedings) is a member of the CASA team, first contact for outside users, and stakeholder representative on behalf of external users. The DPUC is an software subcommittee of the NRAO Users Committee, and consist of ten user representatives that advise NRAO's devision on Data Management ands Systems (DMS) on all things related to software via a yearly DPUC meeting and written report with recommendations.

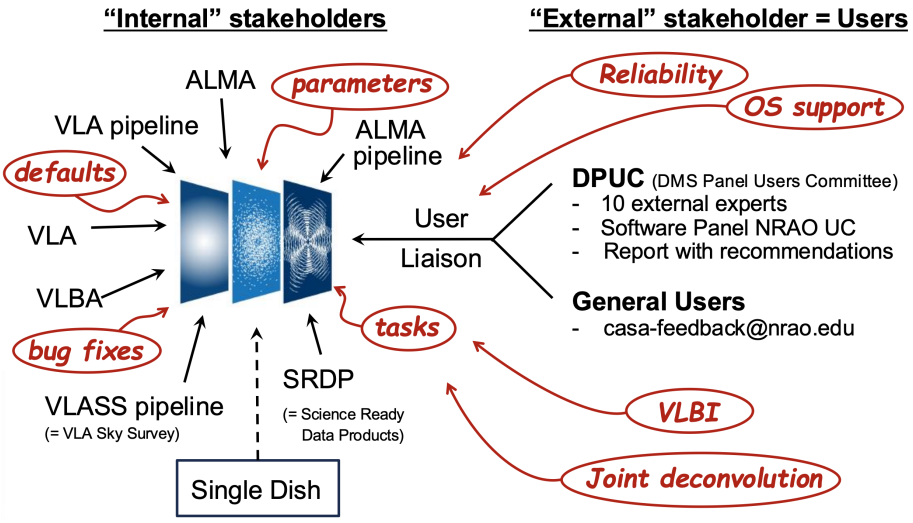


Figure 2. CASA development priorities through stakeholder representation. "Internal" stakeholders represent ALMA, VLA, VLBA, and other groups at NRAO, with Single Dish priorities handled by NAOJ. General users have "external" stakeholder representation through the CASA User Liaison and DMS Panel of the Users Committee. Internal stakeholders often focus on short-term requirements needed for operations, while the user community provides input on long-term requirements that take a 'big-picture' approach to keeping CASA versatile and robust.

Experience shows that requirements from the external user community focus on longer term goal that tackle the "big-picture" approach to keeping CASA a versatile and robust software package. An example has been the desire by the community to make CASA more reliable (Emonts 2018), which triggered the CASA team to implement an extensive and robust testing philosophy and infrastructure (see contribution of S. Castro as part of these proceedings). Other examples are the continuous request to keep support for Mac OS (not used in telescope operations), increased support for VLBI, and the request to offer the possibility for joint deconvolution of single-dish and interferometry data (which resulted in the development a new task `sdintimaging`). The general user community therefore provides a critical role in the healthy, long-term development of the CASA software.

4. User Support

Development teams also need to provide feedback to users regarding the use of their software, and assist when problems arise. User support requires a dedicated investment of a certain full- or fractional-time equivalent (FTE) workload. It is key that software teams balance user support based on available FTEs, to avoid distractions among the development team, and keep focus on maintaining and improving the software. A main challenge in this is to manage user expectations accordingly.

Software teams have many avenues to provide a balanced user support. These include *active support* in the form of personal contact between the software team and users, sometimes via formal platforms, such as helpdesks. The user community may also benefit from *passive support* in the form of tutorials and demos, Newsletters, or an online forum. A third avenue that is essential support for users, and which is almost mandatory for any software team, is what we define here as *technical support*, which includes code documentation, as well as a dedicated suite of tests that run on various Operating Systems (OSs). In the next Sections, we will address how CASA provides some of these critical support structures.

4.1. CASA Feedback: casa-feedback@nrao.edu

The need for balancing user support with resources is often most critical regarding one-on-one user support. A dedicated helpdesk, where users can raise problems and expect swift help to find a resolution, is a platform that is most useful for the user community, but can become a serious constraint for development teams. This is visualized in Fig. 3, where we summarize CASA-specific tickets that where handled by the telescope support teams through the NRAO and ALMA Helpdesk during the period of ten months (Emonts 2018). The majority of incoming tickets were related to misunderstandings by the user, bad data, or issues that had been resolved in more recent CASA releases. Only a minority of tickets revealed a problem with the CASA code or documentation. While it is critical to capture this feedback from the community to ensure that problems with the code are addressed, the workload involved in sorting out software problems from the non-software issues can be significant. Therefore, running an official helpdesk can diminish the often already strained resources of software teams for code development.

A more informal avenue for communication with the user community may be beneficial if FTEs for user support are limited. The CASA team adopted such an approached by introducing a dedicated contact email: casa-feedback@nrao.edu. Casa-feedback@nrao.edu is intended for general feedback to the CASA team, and is also

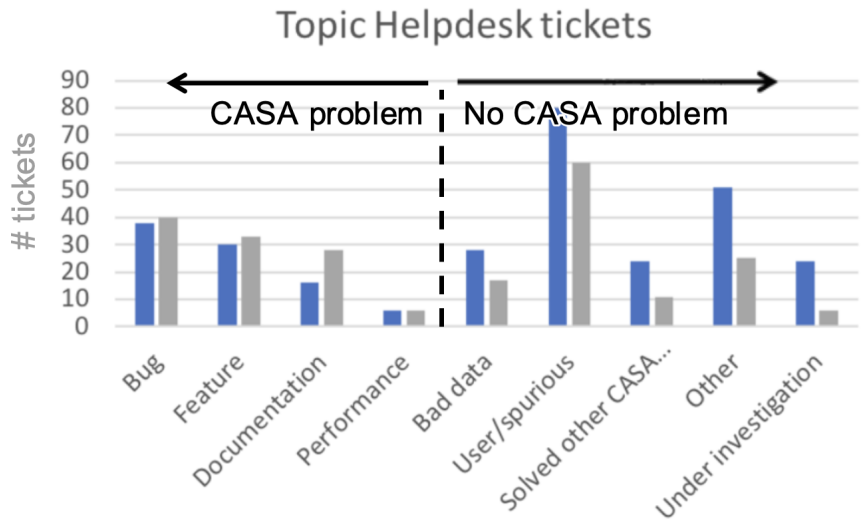


Figure 3. Helpdesk statistics. Total number of CASA-labeled Helpdesk tickets received by the NRAO and ALMA helpdesks during a 10-month period in 2018, grouped by topic. Only a minority of tickets revealed a problem with the CASA code base or documentation. Reproduced from CASA Memo 6 (Emonts 2018).

used for best-effort support for users who encounter problems with the CASA code or documentation. Contrary to a helpdesk system, this dedicated email address does not guarantee one-on-one problem solving or automated archiving of tickets, but it has several advantages that keep the process efficient and manageable:

- *Best-effort:* a best-effort support system minimizes the pressure on development resources, allowing developers to respond to reported issues when the development schedule allows;
- *Direct CASA contact:* the CASA team is not actively involved on the NRAO and ALMA helpdesks, therefore a dedicated email provides a direct means for users (and helpdesk staff) to contact the CASA team;
- *Multiple recipients:* a dedicated email system allows multiple recipients, which means that we have multiple sets of eyes on incoming issues. This increases the response time while minimizing the overheads involved in consulting other team-members;
- *Feedback:* casa-feedback@nrao.edu is also an easy avenue for users to provide general feedback on the CASA software, and on its successor, the Radio Astronomy Data Processing System (RADPS; see Sect. 4).

We welcome users to use casa-feedback@nrao.edu for feedback or questions concerning CASA.

4.2. CASA Documentation

Code documentation is often the first point of contact for users with questions about the software. As such, it is essential that software packages include a complete and accurate documentation that users can read, and be referred to, when encountering problems.

CASA Docs[1] is the official documentation of the CASA sofware (Fig. 4). It is written alongside development of the codebase. In addition, tests to verify that the CASA code is working as intended are written against CASA Docs. This optimizes efforts to ensures that the codebase and documentation are in sync and accurate.

CASA Docs provides extensive descriptions of CASA tasks and their parameters. These task pages can optionally be opened from the CASA command-line by typing `doc('taskname')`. CASA Docs also provides documentation on the tools and GUIs, as well as instructions on how to update the data repository. In addition, it contains general chapter pages with detailed background information on calibration and imaging, Release Information and Known Issues, installation instructions and OS compatibility, CASA Memos and Knowledgebase articles, Community Examples on running CASA in Jupyter Notebooks and Colab, and contact information. A new version of CASA Docs is published with each CASA release.

The CASA software is also described in several recent publications (CASA Team et al. 2022; van Bemmel et al. 2022; Emonts et al. 2025).

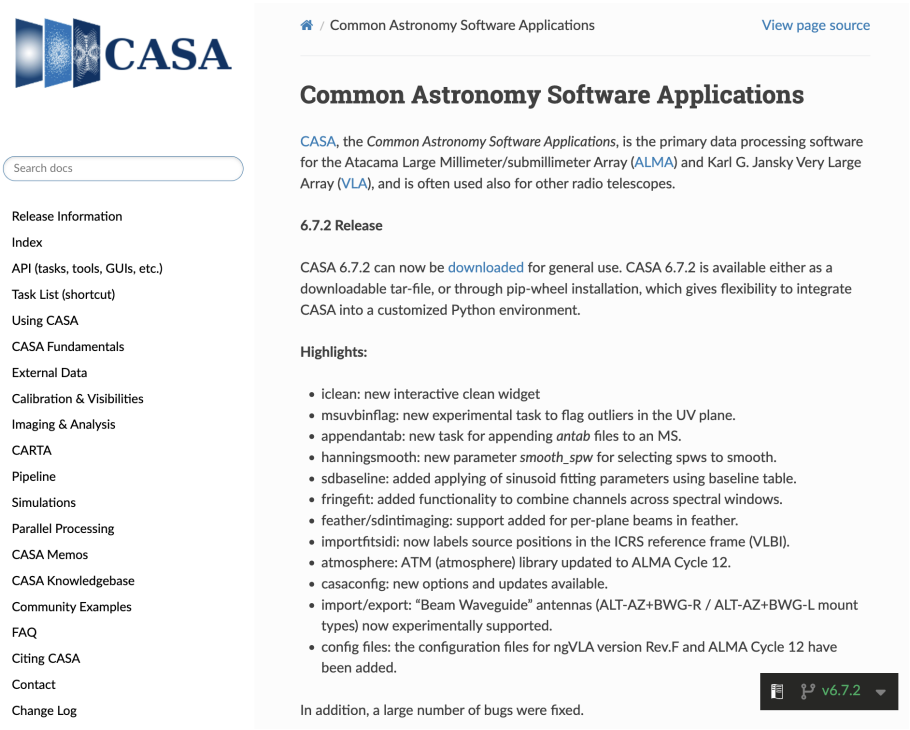


Figure 4. Layout of the CASA Docs documentation. Shown is the landing page for the 6.7.2 release version, with a navigatable list of pages and search box on the left. CASA Docs is hosted on Github and can be accesses through readthedocs: `https://casadocs.readthedocs.io`

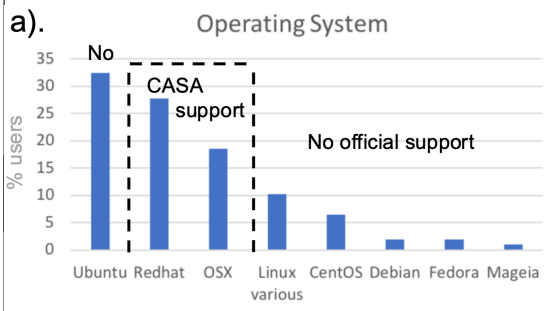
[1] `https://casadocs.readthedocs.io`

4.3. CASA Testing

A large fraction of user questions concerns CASA installation and OS support. Results from a 2018 CASA User Survey revealed that Ubuntu is the most widely used OS for running CASA among the external user community (Fig. 5a ; Emonts 2018). This despite the fact that CASA is only officially supported for the latest releases of Linux RedHat and Mac OS, and only fully validated against the current operational Linux/RedHat configuration of NRAO instruments.

In order to meet user demand regarding the versatility of operating systems, the CASA team incorporated combinations of the most widely used OS and Python versions at various places in the suite of tests that validate the CASA code. The compatibility of CASA with these OS/Python versions is advertised in CASA Docs (Fig. 5b). This approach allows the CASA team to focus efforts on developing the code against the OS/Python version used in ALMA and VLA operations, while providing feedback to the community for which other OS/Python versions CASA is expected to work. Users can report a bug through casa-feedback@nrao.edu if CASA fails on any of the compatible OS/Python versions that are advertised in CASA Docs.

a).



b).

OS	casatools	casatasks	casashell	aux repos	stakeholders	regressions	mpi*
RHEL 8 + Py 3.12	T	T	T	T	T	T	T
RHEL 8 + Py 3.13	T	T					
RHEL 9 + Py 3.12	T	T		T		T	T
RHEL 9 + Py 3.13							
Ubuntu 24 + Py 3.12	T	T		T			
Ubuntu 24 + Py 3.13							
MacOS 15 + Py 3.12	T	T					
MacOS 15 + Py 3.13	T	T					

Figure 5. CASA compatibility with different Operating Systems. *a).* Survey statistics on the use of Operating Systems among the external user community. The figure is reproduced from CASA Memo 6 (Emonts 2018). The majority of users relies on Ubuntu for CASA processing, despite the fact that CASA is only officially supported for Linux/RedHat and Mac OS. *b).* Automated testing table in CASA Docs, where "T" indicates tests that have run and successfully completed. This table is guidance for the community in that it lists all combinations of OS and Python that are expected to be compatible with the corresponding CASA release version.

5. Outlook

CASA is a widely used data-processing software for radio astronomy, with an active role for the user community in defining development requirements, and with efficient use of resources for providing support to the community. While CASA has become the predominant software for radio astronomy, the advancement of radio telescopes, and associated dramatic increase in data rates and volumes, introduces limitations for a code that is two decades old (McMullin et al. 2007).

For the foreseeable future, two major infrastructure projects dominate the requirements for radio data processing at NRAO and partner institutes. These are the ALMA Wideband Sensitivity Upgrade (WSU; Carpenter et al. 2023) and the Next-Generation

Very Large Array (ngVLA; Murphy 2018). Both instruments will provide major challenges for data processing software, resulting from orders of magnitude higher data rates and volumes (e.g., Bhatnagar et al. 2021; Kepley et al. 2024). The current CASA software will not be sufficient for efficiently processing data from these instruments.

As a result, efforts are underway to develop a next-generation Radio Astronomy Data Processing System (RADPS). RADPS will have as primary objective the production of high-level data production from ALMA-WSU and ngVLA, with secondary goal to support the evolution of radio astronomy data processing through a widely accessible package. RADPS will rely on scalable data-processing capabilities using mostly off-the-shelf technology, with intuitive user interfaces.

A detailed discussion of RADPS is beyond the scope of these proceedings. However, RADPS development heavily leans on CASA developers. It is therefore anticipated that development of CASA will continue to wind down in favor of RADPS. This will result in CASA transitioning towards a maintenance mode in the coming years, with an expected sunset of CASA software development in the foreseeable future. Regardless, the legacy of CASA will be one where users played a critical role over the lifetime of the software.

6. CASA Resources

The following is an overview of CASA resources available to the user community:

- **CASA contact:** we welcome feedback from users!
  (casa-feedback@nrao.edu)
- **CASA announcements:** stay up-to-date on CASA announcements and register!
  (`https://listmgr.nrao.edu/mailman/listinfo/casa-announce`)
- **CASA Website:** official CASA website
  (`https://casa.nrao.edu`)
- **CASA Docs:** official CASA documentation, or *"How the code works"*
  (`https://casadocs.readthedocs.io`)
- **CASA Guides:** data reduction tutorials, or *"How to work the code"*
  (`https://casaguides.nrao.edu` – maintained by instrument support teams)
- **CASA Code:** open-source codebase on Github and Bitbucket
  (`https://github.com/casangi`)
  (`https://open-bitbucket.nrao.edu/projects/CASA/repos/casa6/browse`)
- **CASA Newsletter:** CASA news sent to the community
  (`https://science.nrao.edu/enews/casa_012/index.shtml`)
- **CASA Reference Paper:** CASA Team et al. 2022, PASP, 134, 114501
  (`https://iopscience.iop.org/article/10.1088/1538-3873/ac9642`)
- **DMS Panel of the Users Committee (DPUC):** software advisory committee
  (`https://safe.nrao.edu/wiki/bin/view/Software/CASA/CASAUsersCommittee`)

Acknowledgments. We thank the organizers of ADASS XXXV for a great conference. We are indebted to the current and past members of the DMS Panel of the Users Committee (formerly the CASA users committee), the user community, and the CASA stakeholders for continuously providing invaluable feedback that drives CASA development. We also thank our colleagues in the pipeline development teams and instrument support groups for their valuable efforts regarding code validation, quality control, and user support. CASA is being developed by an international team of scientists and software engineers based at the National Radio Astronomy Observatory (NRAO), the European Southern Observatory (ESO), the National Astronomical Observatory of Japan (NAOJ), and the Joint Institute for Very Long Baseline Interferometry European Research Consortium (JIVE), under the guidance of NRAO. The National Radio Astronomy Observatory is a facility of the National Science Foundation operated under cooperative agreement by Associated Universities, Inc. ALMA is a partnership of ESO (representing its member states), NSF (USA) and NINS (Japan), together with NRC (Canada), MOST and ASIAA (Taiwan), and KASI (Republic of Korea), in cooperation with the Republic of Chile. The Joint ALMA Observatory is operated by ESO, AUI/NRAO and NAOJ.

References

Astro2020: National Academies of Sciences, Engineering, and Medicine 2023, Pathways to Discovery in Astronomy and Astrophysics for the 2020s (Washington, DC: The National Academies Press). URL https://nap.nationalacademies.org/catalog/26141/pathways-to-discovery-in-astronomy-and-astrophysics-for-the-2020s

Bhatnagar, S., Hiriart, R., & Pokorny, M. 2021, Size-of-Computing Estimates for ngVLA Synthesis Imaging, ngVLA Computing Memo #4, August 2021, 22 pages

Carpenter, J., Brogan, C., Iono, D., & Mroczkowski, T. 2023, in Physics and Chemistry of Star Formation: The Dynamical ISM Across Time and Spatial Scales, edited by V. Ossenkopf-Okada, R. Schaaf, I. Breloy, & J. Stutzki, 304. eprint: 2211.00195

CASA Team, et al. 2022, PASP, 134, 114501. eprint: 2210.02276

Casacore Team 2019, casacore: Suite of C++ libraries for radio astronomy data processing, Astrophysics Source Code Library, record ascl:1912.002. eprint: 1912.002

Emonts 2018, CASA Memo 6: User Survey and Helpdesk Statistics, CASA Docs Memo Series. URL https://casadocs.readthedocs.io/en/stable/notebooks/memo-series.html

Emonts, B., van Bemmel, I., Moellenbrock, G., Kettenis, M., Small, D., Rau, U., the CASA Team, & the CASA-VLBI Team 2025, in Astronomical Data Analysis Software and Systems XXXII, edited by S. Gaudet, D. Bohlender, S. Gwyn, A. Hincks, & P. Teuben, vol. 538 of Astronomical Society of the Pacific Conference Series, 154

Hunter, T. R., et al. 2023, PASP, 135, 074501. eprint: 2306.07420

Kepley, A. A., Brogan, C. L., Carpenter, J., Diaz Trigo, M., Hatsukade, B., & Antognini, J. 2024, Estimates of ALMA WSU Data Properties, ALMA Memo Series, 626, January 31, 2024, 40 pages

McMullin, J. P., Waters, B., Schiebel, D., Young, W., & Golap, K. 2007, in Astronomical Data Analysis Software and Systems XVI, edited by R. A. Shaw, F. Hill, & D. J. Bell, vol. 376 of Astronomical Society of the Pacific Conference Series, 127

Murphy, E. (ed.) 2018, Science with a Next Generation Very Large Array, vol. 517 of Astronomical Society of the Pacific Conference Series

Perez, F., & Granger, B. E. 2007, Computing in Science and Engineering, 9, 21

Shortridge, K. 2001, in Organizations and Strategies in Astronomy, Volume II, edited by A. Heck, vol. 266 of Astrophysics and Space Science Library, 163

van Bemmel, I. M., et al. 2022, PASP, 134, 114502. eprint: 2210.02275

Astronomical Data Analysis Software and Systems XXXV
*ASP Conference Series, Vol. **VOL** Compiled: Tuesday 30th June, 2026@ 01:30*
K. Polsterer, D. Bohlender, S. Gwyn, A. Hincks and P. Teuben, eds.
*© **YEAR** Astronomical Society of the Pacific*

ZZZDEBUG!045.tex
Stenborg, T. N.|textbf
applications!DS9
data formats!FITS
software!Linux
computer languages!shell script
software!macOS
DS9! ascl:0003.002
IRAF! ascl:9911.002

DS9 FITS Segmentation with Linux and macOS Shell Script Metaprogramming

T. N. Stenborg

The University of Sydney, Sydney, New South Wales 2006, Australia;
tstenbor@acm.org

Abstract. Optical survey imaging in FITS files can exceed the screen size of image viewers. To inspect such data, users can either traverse the image or zoom out. Manual traversal risks inefficient or incomplete image inspection, subject to the available image viewer controls. Zooming out compromises image detail. A solution to these drawbacks is segmenting images to fit within a user's screen, allowing inspection in slideshow fashion. Shell script metaprogramming SAOImageDS9 to perform FITS segmentation is presented. The solution infrastructure is simple, needing only DS9 and a Linux or macOS shell to run.

Emission line objects can be discovered in optical imaging by combining narrowband and broadband data. Such combination might be subtractive, for difference imaging, or additive via different color channels of an image viewer. Interleaving different combinations for blink comparison was applied to planetary nebula candidate identification, demonstrating the scripts' utility in semi-automating astronomical discovery. Associated resources are available publicly via GitHub.

1. Introduction

A 2012–2016 project uncovered a new population of candidate planetary nebulas (PN) in [O III] imaging toward the Galactic bulge (Stenborg & Parker 2014; Stenborg 2016, 2017). It used imaging from the Blanco 4 m telescope, Cerro Tololo Inter-American Observatory (CTIO), Chile (Leverington 2013, 2016). Specifically, Mosaic II CCD imaging (Muller et al. 1998), with [O III] and off-band filters (serial numbers c6014, k1015),[1] collected and reduced in a third-party PhD study (Kovacevic 2011).

SAOImageDS9 (hereafter, DS9) (Joye & Mandel 2003, 2004, 2005; Joye 2006, 2011) was used to review Mosaic II FITS images for that bulge PN project. Mosaic II's field of view was $36' \times 36'$ (Schmidt 2006). If images were zoomed in to provide enough detail to distinguish candidate PNs, systematically navigating images using *pan* mode was difficult. DS9 shell script metaprogramming was used to segment images into frames smaller than the user's screen size, allowing review, slideshow fashion, in external image viewers (Stenborg 2014, 2016). Accidentally missing image sections during review was thus avoided. Frames were also configured for areal overlap, to help prevent missing details at frame edges.

This work describes a refined, more robust, better documented, multi-platform validated version of the DS9 FITS segmentation metaprogramming.

[1] NOIRLab filter specifications https://noirlab.edu/science/filters

2. Methods

2.1. Shell Script Metaprogrammed FITS Segmentation

A Bash shell script was composed to metaprogram, and execute, DS9 command line segmentation of FITS files. The script opens and pans over FITS files in DS9, exporting JPEG snapshots at ~80% screen size, until each file has been entirely segmented into images. Using a subset of full screen size allowed for systems reserving screen display areas for desktop elements, such as menus and taskbars. The script's segmentation algorithm also allowed for heterogeneity across operating systems in the rendered size of DS9's own GUI elements, such as its menu and title bars, window borders, etc.

A high-level description of the segmentation is given in Algorithm 1. The segmentation script was validated using the ShellCheck syntactic linter (Lazarek et al. 2025).

algorithm FITS_File_Segmentation

input *fileList, widthFITS, heightFITS, widthDisplay, heightDisplay*

forall *file* **in** *fileList*
  $x_{\max} := \lceil widthFITS / widthDisplay \rceil - 1$
  $x_{\mathrm{pan}_1} := x_{\max} + 1$
  $x_{\mathrm{pan}_2} := x_{\max} / (2 \times x_{\mathrm{pan}_1})$
  $y_{\max} := \lceil heightFITS / heightDisplay \rceil - 1$
  $y_{\mathrm{pan}_1} := y_{\max} + 1$
  $y_{\mathrm{pan}_2} := y_{\max} / (2 \times y_{\mathrm{pan}_1})$
  for $x = 0$ **to** $x_{\max}$ **do**
    $panHorz = \lfloor widthFITS \times (x / x_{\mathrm{pan}_1} - x_{\mathrm{pan}_2}) \rceil$
    for $y = 0$ **to** $y_{\max}$ **do**
      $panVert = \lfloor heightFITS \times (y / y_{\mathrm{pan}_1} - y_{\mathrm{pan}_2}) \rceil$
      DS9_Save_Image(*file*, *panHorz*, *panVert*)
    end for
  end for
end forall
exit

Algorithm 1. High-level FITS file segmentation algorithm, using ceiling $\lceil x \rceil$ and rounding $\lfloor x \rceil$ operations. Shell script metaprogrammed DS9 commands pan horizontally and vertically, saving JPEG images displayed at ~80% screen size, for arbitrary FITS file and screen dimensions.

The script has two operating modes. The first is *single* mode, intended for processing FITS files in single DS9 frames. The second is *dual* mode, designed to process pairs of complementary FITS files in the red and green channels of RGB DS9 frames. The operating modes are invoked via command line options. Details, including example syntax, may be displayed via a `-h` help option, i.e., `./fits_segmenter.sh -h` at a command line.

The script's output naming scheme was designed for interleaving related data on a file system, provided the input FITS files have a common file prefix. This feature was used in the 2012–2016 bulge PN project to process [O III] and off-band filter data in *dual* frame (RGB) mode, then difference imaging in *single* frame mode. This in turn enabled blink comparison of segmented FITS snapshots in image viewers such as Preview on the macOS, for efficient PN candidate discovery. [O III]-bright PNs were prominent red in RGB frames, and faint PNs were discernible in difference imaging.

2.2. Galactic Bulge Test Imaging Data

Since the 2012–2016 bulge PN project, raw and processed data products from associated observations (programs 2008A-0549, 2009A-0405) have become freely available from the NOIRLab Astro Data Archive.[2] This work used a subset of public domain, pre-stacked, Mosaic II imaging from program 2008A-0549 as test data. Fields F1759-2841, F1759-2915 and F1803-2807 were used (Kovacevic 2011; Stenborg 2016).

Corresponding difference imaging was produced by subtracting off-band frames from [O III] on-band frames with IRAF (Tody 1986, 1993), community distribution 2.17.1, on Ubuntu 24.04.3. Difference subtrahends, for each on/off-band frame pair, were off-band pixel intensities downscaled by the ratio of the median off-band to on-band frame intensities.

3. Results and Discussion

The segmentation shell script was successfully executed against the test bulge imaging data (see e.g., Figure 1). System configurations validated are given in Table 1.

Table 1. Configurations confirmed compatible with the segmentation shell script.

Operating System	DS9 Version
Debian GNU/Linux 13	8.7b2 (Debian 12 Intel)
Kali GNU/Linux Rolling (2025.3)	8.7b2 (Debian 12 Intel)
macOS Monterey 12.7.6	8.6 (macOS Monterey 12 Intel, X11 port)
openSUSE Leap 16.0	8.7b2 (OpenSUSE 15 Intel)
Ubuntu 24.04.3 LTS	8.7b2 (Ubuntu 24 Intel)

A limitation of the shell script is its design for single-monitor systems. Note too that for the macOS, because metaprogramming DS9 commands is a key component of the shell script, the X11 port of DS9 is needed, not the Aqua port.

A strength of the shell script is its low-infrastructure requirements. It needs only DS9, a Bash shell, and some OS-level commands and command line utilities.

The DS9 FITS segmenter is ready for community use. The shell script, IRAF reduction script and documentation are all available on GitHub.[3]

Acknowledgments. This work used data obtained from the US National Science Foundation's (NSF) NOIRLab Astro Data Archive. The data are associated with observations at the NSF CTIO, NSF NOIRLab (NOIRLab Prop. ID 2008A-0549; PI: Q. Parker), which is managed by the Association of Universities for Research in Astronomy (AURA) under a cooperative agreement with the US NSF. Data reduction was performed with Community IRAF. Use was also made of SAOImageDS9, developed by the Smithsonian Astrophysical Observatory.

[2]NOIRLab Astro Data Archive `https://astroarchive.noirlab.edu`

[3]GitHub repository `https://github.com/tstenborg/DS9-FITS-Segmentation`

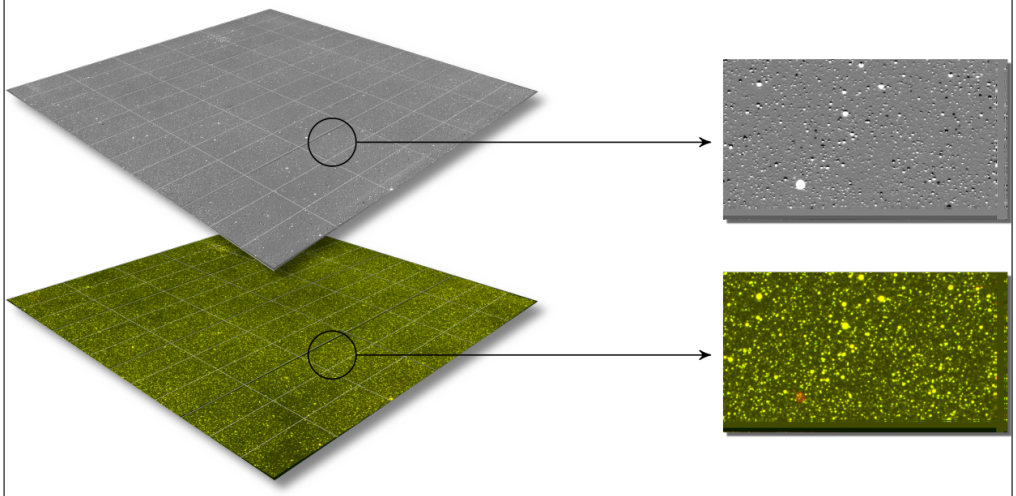


Figure 1. Example [O III] on/off band (bottom) and difference (top) image segments for CTIO field F1803-2807, holding PN G002.6-03.1.

References

Joye, W. 2011, in ADASS XX, edited by I. N. Evans, A. Accomazzi, D. J. Mink, & A. H. Rots, vol. 442 of ASP Conf. Ser., 633

Joye, W. A. 2006, in ADASS XV, edited by C. Gabriel, C. Arviset, D. Ponz, & S. Enrique, vol. 351 of ASP Conf. Ser., 574

Joye, W. A., & Mandel, E. 2003, in ADASS XII, edited by H. E. Payne, R. I. Jedrzejewski, & R. N. Hook, vol. 295 of ASP Conf. Ser., 489

— 2004, in ADASS XIII, edited by F. Ochsenbein, M. G. Allen, & D. Egret, vol. 314 of ASP Conf. Ser., 505

— 2005, in ADASS XIV, edited by P. Shopbell, M. Britton, & R. Ebert, vol. 347 of ASP Conf. Ser., 110

Kovacevic, A. 2011, Ph.D. thesis, Macquarie University, Dept. Physics and Astronomy

Lazarek, L., et al. 2025, in Proc. 2025 Workshop on Hot Topics in Operating Systems, HotOS '25, 88

Leverington, D. 2013, Encyclopedia of the History of Astronomy and Astrophysics (Cambridge)

— 2016, Observatories and Telescopes of Modern Times (Cambridge)

Muller, G. P., Reed, R., Armandroff, T., Boroson, T. A., & Jacoby, G. H. 1998, in Optical Astronomical Instrumentation, edited by S. D'Odorico, vol. 3355 of SPIE Conf. Ser., 577

Schmidt, R. E. 2006, vol. 336 of Astrophysics and Space Science Library, 353

Stenborg, T. N. 2014, in Asymmetrical Planetary Nebulae VI Conference, edited by C. Morisset, G. Delgado-Inglada, & S. Torres-Peimbert, 99

— 2016, Ph.D. thesis, Macquarie University, Dept. Physics and Astronomy

— 2017, PASP, 129, 067002

Stenborg, T. N., & Parker, Q. A. 2014, in Asymmetrical Planetary Nebulae VI Conference, edited by C. Morisset, G. Delgado-Inglada, & S. Torres-Peimbert, 98

Tody, D. 1986, in Instrumentation in Astronomy VI, edited by D. L. Crawford, vol. 627 of SPIE Conf. Ser., 733

— 1993, in ADASS II, edited by R. J. Hanisch, R. J. V. Brissenden, & J. Barnes, vol. 52 of ASP Conf. Ser., 173

Astronomical Data Analysis Software and Systems XXXV
*ASP Conference Series, Vol. **VOL** Compiled: Tuesday 30th June, 2026@ 01:30*
K. Polsterer, D. Bohlender, S. Gwyn, A. Hincks and P. Teuben, eds.
*© **YEAR** Astronomical Society of the Pacific*

ZZZDEBUG!047.tex
Zhang, B.|textbf

CSST Data Processing Pipeline: Architecture & Development

Bo Zhang[1]

[1]*National Astronomical Observatories, Chinese Academy of Sciences, Beijing 100101, People's Republic of China;* bozhang@nao.cas.cn

Abstract. The Chinese Space Station Survey Telescope (CSST) mission is expected to generate approximately 10 PB of data over its 10-year operational lifetime, presenting significant data processing challenges. To address these challenges, we have designed and implemented the *CSST Data Processing Pipeline* based on a cloud-native architecture. The pipeline leverages Kubernetes as the underlying orchestration platform, providing scalability, resilience, and efficient resource management essential for petabyte-scale astronomical data processing. The system is structured into three core components: a polyglot-persistence database layer for managing heterogeneous data products, a workflow orchestration layer utilizing Argo Workflows for containerized data processing operators, and a client layer providing user interfaces. The development process follows Agile and DevOps principles, incorporating a robust CI/CD pipeline implemented with Jenkins to ensure systematic integration, testing, and deployment of code updates. This cloud-native approach demonstrates significant advantages in automation, maintainability, and resource efficiency, providing a solid foundation for handling the massive data flow from the CSST mission. The successful implementation of this pipeline will be critical in maximizing the scientific return of the CSST, enabling groundbreaking discoveries in astronomy.

1. Introduction

The Chinese Space Station Survey Telescope (CSST, Zhan 2011), the leading scientific payload of the Tiangong Space Station, features a 2-meter aperture and an off-axis three-mirror anastigmat (TMA) optical design. The telescope is engineered to operate in a co-orbital trajectory with the space station, enabling periodic docking for servicing, replenishment, and potential instrument upgrades. This flagship mission has a nominal operational lifetime of 10 years and is projected to launch in 2027.

The CSST is equipped with five advanced scientific instruments mounted on its focal plane: the Main Survey Camera (MSC), the Multi-Channel Imager (MCI), the Integral Field Spectrograph (IFS), the Cool Planet Imaging Coronagraph (CPIC), and the High-Sensitivity Terahertz Detection Module (HSTDM). Observation time is strategically allocated, with approximately 70% dedicated to the MSC for wide-field surveys, while the remaining four instruments collectively share 20% for targeted observations. The final 10% of the timeline is reserved for on-orbit servicing. After the 10-year operational period, the wide-field survey will cover 17,500 deg^2 with multi-band imaging in 7 filters (NUV, u, g, r, i, z, y, to ~25.5 mag) and slit-less spectra (GU, GV, GI, to ~23.2 mag, with $R \geq 200$), which is expected to significantly advance our understanding of

cosmology, galaxies, the Milky Way, and stars (see CSST Collaboration et al. 2026, for more details).

The daily data downlink volume is approximately 2 TB, with derived data products exceeding ten times that amount. Over the 10-year survey, the total data volume is expected to reach 10 PB. To address the resulting large-scale data processing challenges, we have designed the *CSST Data Processing Pipeline* based on cloud computing technologies, employing CI/CD tools to improve development efficiency. This paper provides a brief introduction to the pipeline architecture.

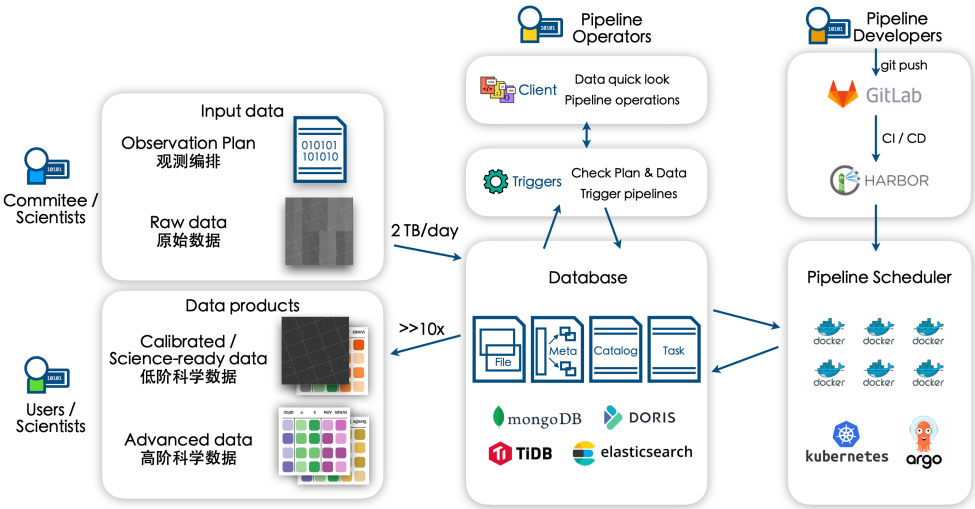


Figure 1. The outline of the architecture of CSST data processing pipeline.

2. Architecture

The *CSST Data Processing Pipeline* is designed with a cloud-native architecture, using Kubernetes (K8s) as the underlying orchestration platform. This approach provides scalability, resilience, and efficient resource management, which are essential for processing the petabyte-scale data products expected from the CSST mission. The entire software system is structured into three core components: the Database, the Workflow Orchestration, and the Client.

2.1. Database

The database serves as the foundational component of the pipeline, responsible for storing and managing heterogeneous data products. These include raw and calibrated files, their associated metadata, astronomical source catalogs, and workflow task information. To address the diverse data access patterns – such as high-throughput metadata queries and large-scale catalog cross-matching – the system employs a polyglot persistence strategy, integrating MongoDB, Doris, TiDB, and Elasticsearch. This integration supports flexible metadata management, efficient full-text search, and high-

performance handling of catalog data encompassing up to 10^{10} astronomical sources, enabling complex spatial queries and data retrieval at scale.

2.2. Workflows

Automated data processing is implemented through a workflow orchestration component. Argo Workflows is chosen as the primary engine due to its native integration with Kubernetes. It allows each data processing step – such as instrumental signature removal, astrometric calibration, and photometric measurement – to be encapsulated into a containerized, reusable Operator. Complex processing sequences are defined as Directed Acyclic Graphs (DAGs), which Argo executes on the Kubernetes cluster, managing dependencies and ensuring fault-tolerant execution. This design supports high-throughput processing by dynamically scaling computational resources based on the data queue.

2.3. Client

The client component provides interfaces for user interaction with the pipeline. It currently consists of two main parts:

- **Triggering Service**: This component is responsible for initiating the execution of predefined DAGs based on specific events, such as the arrival of new data (group) or a scheduled task.

- **Command-Line Tool (CLI)**: A CLI offers astronomers and pipeline operators direct control for submitting jobs, monitoring workflow status, and managing data processing tasks.

A web-based Graphical User Interface (GUI) is under active development. This interface will offer visual workflow monitoring, interactive data product quick look / exploration, and simplified job submission capabilities, making the system more accessible.

3. Development

The development of the pipeline follows Agile and DevOps principles. The codebase is managed using Git for version control, and a robust *Continuous Integration and Continuous Deployment* (CI/CD) pipeline is implemented using Jenkins. An automated workflow is triggered upon new code pushes to the GitLab repository. Jenkins then executes a comprehensive suite of unit tests and, upon successful completion, automates branch merging. The integrated code is subsequently packaged into Docker images and published to a centralized Docker registry, making the updated images ready for deployment in the Kubernetes cluster. This end-to-end automation guarantees that every code change is systematically integrated, tested, and deployed, substantially improving development efficiency and system reliability. In addition, the modular design of processing operators combined with standard containerization enhances collaborative development, facilitates independent testing of algorithms, and promotes long-term maintainability.

4. Conclusion and prospects

The cloud-native *CSST Data Processing Pipeline* presents a robust solution for petabyte-scale astronomical data processing through its Kubernetes-based architecture. By integrating a polyglot database, Argo Workflows, and a versatile client component, the system ensures high efficiency in automation and resource management, underpinned by DevOps practices .

Future development will focus on enhancing the web-based GUI for improved user interaction and integrating AI techniques to elevate the quality and precision of data products. These efforts are pivotal to transforming the pipeline into an intelligent discovery engine, thereby maximizing the scientific return of the CSST mission.

Acknowledgments. This work is supported by the China Manned Space Project via the CSST scientific data processing and analysis system and grant No. CMS-CSST-2025-A21.

References

CSST Collaboration, Gong, Y., Miao, H., Zhan, H., Li, Z.-Y., Shangguan, J., Li, H., Liu, C., Chen, X., Yuan, H., Zhou, J., Liu, H.-G., Yu, C., Ji, J., Qi, Z., Liu, J., Dai, Z., Wang, X., Zheng, Z., Hao, L., Dou, J., Ao, Y., Lin, Z., Zhang, K., Wang, W., Sun, G., Li, R., Li, G., Xu, Y., Li, X., Li, S., Wu, P., Zhang, J., Wang, B., Bai, J., Cai, Y.-F., Cai, Z., Cao, J., Chan, K. C., Chang, J., Chen, X., Chen, X., Chen, Y., Chen, Y., Cui, W., Dong, S., Du, P., Duan, W., Fan, J., Fan, L., Fan, Z., Fan, Z., Fang, T., Fu, J., Fu, L., Fu, Z., Gao, J., Gu, S., Gu, Y., Guo, Q., Han, Z., Hu, B., Huang, Z., Ho, L. C., Jiang, L., Jiang, N., Jing, Y., Kang, X., Kong, X., Li, C., Li, C., Li, D., Li, J., Li, N., Li, Y. A., Liao, S., Lin, W., Liu, F., Liu, J., Liu, X., Liu, Z., Mao, R., Mao, S., Meng, X., Pang, X., Peng, X., Peng, Y., Shan, H., Shen, J., Shen, S., Shen, Z., Shi, S.-C., Shi, Y., Tan, S., Tian, H., Wang, J., Wang, J.-X., Wang, X., Wang, Y., Wu, H., Wu, J., Wu, X., Xu, C., Xue, X.-X., Xue, Y., Yang, J., Yang, X., Yao, Q., Yuan, F., Yuan, Z., Zhang, J., Zhang, P., Zhang, T., Zhang, W., Zhang, X., Zhao, G., Zhao, G., Zhong, H., Zhong, J., Zhou, L., Zhu, W., & Zu, Y. 2026, Science China Physics, Mechanics, and Astronomy, 69, 239501. eprint: 2507.04618

Zhan, H. 2011, Scientia Sinica Physica, Mechanica & Astronomica, 41, 1441

Astronomical Data Analysis Software and Systems XXXV
*ASP Conference Series, Vol. **VOL** Compiled: Tuesday 30th June, 2026@ 01:30*
K. Polsterer, D. Bohlender, S. Gwyn, A. Hincks and P. Teuben, eds.
*© **YEAR** Astronomical Society of the Pacific*

ZZZDEBUG!048.tex
Illingworth, M.|textbf
O'Brien, B.
Hambly, N.
Harnqvist, S.
Utrilla Molina, E.

Evolving the Gaia Data Mining Platform to Near-petabyte scale

Malcolm Illingworth,[1] Brendan O'Brien,[1] Nigel Hambly,[1] Simon Harnqvist,[1] and Enrique Utrilla Mollina,[2]

[1]*Institute For Astronomy, University Of Edinburgh, Edinburgh, United Kingdom;*

[2]*European Space Agency, Madrid, Spain*

Abstract. The forthcoming Gaia Data Release 4 will increase the Gaia mission dataset from 7TB to 600 TB, and to petabyte scale for the final Data Release 5. We describe our progress in building a notebook-based data mining platform to support the current data release 3, and our progress towards enabling the platform to support datasets at petabyte size.

1. Gaia Mission Overview and Data Releases

Gaia is an ESA optical space astronomy mission that was launched on 19th December 2013 and passivated in April 2025, with an observing period of 10.5 years from mid-July 2014 to mid-January 2025. During the mission, approximately 2.1 billion individual sources were observed, with on average 60 observations per source. The datasets comprise a combined astrometric, photometric and spectroscopic survey. The survey contains precise positions, tangential motions, distances; low dispersion spectrophotometry; high dispersion spectroscopy for radial velocities. Sources in the survey are mainly stars, but also millions of asteroids, galaxies and quasars. The Gaia datasets are released periodically, approximately every 2 years. For each release, the entire dataset is reprocessed to increase accuracy and number of observations per source, leading to a near exponential growth in released dataset size. This increase is driven by the increased number of observations per source rather than a dramatic increase in the number of individual sources.[1]

2. Data Mining Platform motivation

Traditional access modes do not scale when datasets reach hundreds of terabytes and beyond, where data download times and provisioning sufficient computing resource may be become logistically impossible outside of specialist high performance computing or cloud environments. The Gaia Data Mining Platform (Gaia DMP)[2] provides a

[1]Gaia Mission Homepage https://www.esa.int/Science_Exploration/Space_Science/Gaia

[2]Gaia Data Mining Platform https://www.gaia.ac.uk/data-releases/mining-platform/the-uk-gaia-data-mining-platform/

code-to-data platform for accessing the current Gaia DR3 dataset via a notebook interface and Spark / Hadoop, targeting applications that need to make use of deep trawls through the dataset. Typical queries include higher order statistical aggregates; pattern searches / matches; wholesale dataset trawling; machine learning applications, such as classification and decision trees. For example, we can run a spectral twins search, where we search the entire DR3 catalogue for spectra matching a chosen source, taking full account of variance / covariance in calibrated mean spectra. This type of search would be extremely difficult to do without a facility such as the Gaia DMP.

3. Initial DMP Prototype

The initial prototype of Gaia DMP to support DR3 consisted of an interactive shared Zeppelin notebook / Hadoop environment. This ran on 1x 54 vcpu shared Zeppelin node and 6x 26 vcpu Hadoop workers, provisioned on OpenStack VMs in the Arcus OpenStack cloud at the University of Cambridge. Installation was via custom Ansible scripts, which also provisioned the underlying OpenStack VMs and networking / storage infrastructure. The DR3 dataset was made available in the environment via Spark. To support more efficient querying via Spark, the DR3 dataset was repartioned and stored as Parquet files on CEPHFS networked storage. This environment worked well in terms of performance and has been stable, however there were a number of issues. The deployment process was complex and not easily replicated in other environments. The infrastructure and application deployment were intermixed, leading to further concerns over portability. With the near 100x increase in the dataset size for DR4, a more scaleable and portable solution was needed. In addition, adding new applications such as Dask was difficult.

4. New Platform

4.1. Cluster Provision

To best support Gaia DR4, we changed the architecture to explicitly split between provision and application. We now deploy Kubernetes clusters on top of OpenStack via ClusterAPI[3]. Currently we have deployments at Edinburgh University (Somerville), Cambridge University (Arcus) and the University Of Barcelona (nCloud2). On Somerville we have access to high memory nodes with fast attached SSD, with Spark configured to use 6TB of fast local SSD cache. On Arcus, we have access to 2 very large memory nodes, with 188 vcpu and 1.9 TB RAM per node. We have also, via the SPACIOUS[4] project, access to Google Cloud Platform and have deployed a cluster there. However, for GCP we deployed using Google's own tools rather than ClusterAPI.

[3] ClusterAPI `https://cluster-api-openstack.sigs.k8s.io/introduction`

[4] SPACIOUS `https://spacious.ub.edu/`

4.2. Astroflow

The Gaia DMP application software (now known as Astroflow[5]) has been completely separated from the cluster provisioning software, and now assumes an existing kubernetes cluster into which to deploy. Each site that we deploy to has different characteristics in terms of cluster capability, storage provision and networking. To help to accomodate these differences, we use a templated deploy using Kustomise. By moving to a cloud native pattern we are more easily able to support new software and scientific applications. Currently we support Jupyter notebooks, allowing Spark and Dask clusters to be created on demand from within user notebooks. We use Keycloak[6] for authentication, allowing integration (if required) with institutional identity providers. User data is managed within the cluster using a virtual filesystem, JuiceFS[7], which allows easy migration between the clusters. Scientific applications currently available within the cluster include: CosmoSIS[8] (Zuntz et al. 2015); CLOE (Euclid Collaboration 2025); GUASOM (Manteiga et al. 2015); GANDALF; Gaia Archive Visualisation Service (Moitinho et al. 2017) and a growing list of future applications.

5. Future Work

We expect to receive DR4 as region-partitioned parquet files, organised as multilevel HEALPix tiles, and should not require any additional reformating for use in the Gaia DMP. Efficient relational joins between datatypes organised by source and observation keys will be transparently supported while spatial cross-match at arbitrary scale is now feasible given the expected spatial organisation in DR4, but will require some additional application code development. Due to the expected large number of parquet files, we are experimenting with providing access to the data via Spark with Apache Iceberg[9]. Apache Iceberg uses an external metadata catalogue to make queries more efficient and should reduce the number of full table scans / parquet files accessed per query. We hope that by supporting Apache Iceberg, supporting datasets from other missions within Gaia DMP should be possible. With the huge growth in the mission dataset sizes, some scientific applications are likely to have very long runtimes. Although we aim to primarily provide an interactive notebook environment, we intend to investigate additionally providing batch mode access for very long running queries and applications.

Acknowledgments. We gratefully acknowledge help and support from all our Gaia DPAC colleagues in developing this service. Computing facilities for development and hosting are provisioned via the UK e-Infrastructure for Research and Innovation at the Science and Technology Facilities Council (the IRIS project). We thank IRIS for generous allocations of computing resources and we are grateful to the University

[5] Astroflow `https://portal.aquarius.astro-flow.com/`

[6] Keycloak `https://www.keycloak.org/`

[7] JuiceFS `https://juicefs.com/en/`

[8] CosmoSIS `https://cosmosis.readthedocs.io/en/latest/`

[9] Apache Iceberg `https://iceberg.apache.org/`

of Cambridge Arcus and University of Edinburgh Somerville OpenStack Cloud services for providing them. Staff funding for the development and operation of the UK Gaia Data Mining Platform is provided by the UK Science and Technology Facilities Council and by the SPACIOUS HORIZON-CL4-2023-SPACE-01-71 project funded under grant agreement no. 101135205 in the UK via the UKRI Horizon guarantee programme.

References

Euclid Collaboration 2025, arXiv e-prints, arXiv:2510.09118. eprint: 2510.09118

Manteiga, M., Dafonte, J. C., Ulla, A., Alvarez, M. A., Garabato, D., & Fustes, D. 2015, in IAU General Assembly, vol. 29, 2254886

Moitinho, A., Krone-Martins, A., Savietto, H., Barros, M., Barata, C., Falcão, A. J., Fernandes, T., Alves, J., Silva, A. F., Gomes, M., Bakker, J., Brown, A. G. A., González-Núñez, J., Gracia-Abril, G., Gutiérrez-Sánchez, R., Hernández, J., Jordan, S., Luri, X., Merin, B., Mignard, F., Mora, A., Navarro, V., O'Mullane, W., Sagristà Sellés, T., Salgado, J., Segovia, J. C., Utrilla, E., Arenou, F., de Bruijne, J. H. J., Jansen, F., McCaughrean, M., O'Flaherty, K. S., Taylor, M. B., & Vallenari, A. 2017, A&A, 605, A52. eprint: 1708.00195

Zuntz, J., Paterno, M., Jennings, E., Rudd, D., Manzotti, A., Dodelson, S., Bridle, S., Sehrish, S., & Kowalkowski, J. 2015, Astronomy and Computing, 12, 45. eprint: 1409.3409

Astronomical Data Analysis Software and Systems XXXV
*ASP Conference Series, Vol. **VOL** Compiled: Tuesday 30th June, 2026@ 01:30*
K. Polsterer, D. Bohlender, S. Gwyn, A. Hincks and P. Teuben, eds.
*© **YEAR** Astronomical Society of the Pacific*

ZZZDEBUG!053.tex
Spreeuw, H.|textbf
Ali, S.
Millenaar, T.
algorithm!source finding
astronomy!transients!scien
alert
images!processing!source
extraction
images!processing!photomet
software!image
processing
data!pipelines!processing
PySE! ascl:1805.026

PySE's new "sky-eater" mode: source extraction for very large images with over 100,000 sources

Hanno Spreeuw,[1] Suvayu Ali,[1] and Timo Millenaar[2]

[1]*The Netherlands eScience Center, Amsterdam, The Netherlands;* h.spreeuw@esciencecenter.nl

[2]*ASTRON (Netherlands Institute for Radio Astronomy), Dwingeloo, The Netherlands*

Abstract. PySE (Python Source Extractor) was developed by Spreeuw & Swinbank between 2005 and 2013 as part of the LOFAR Transients Key Project. It was in continuous use as part of the Amsterdam–ASTRON Radio Transients Facility and Analysis Center (AARTFAAC) pipeline.

Near-real-time performance is scientifically critical for transient surveys, which may generate a new sky image every second. If source extraction increasingly lags behind data acquisition, timely follow-up observations may become impossible.

To meet these requirements, we present the new "sky-eater" mode of PySE, which enables rapid processing of very large images containing large numbers of sources. In this mode all sources from a representative Square Kilometre Array (SKA) artificial test image of 4096×4096 pixels, containing 167,281 inserted sources, are measured within 0.9 seconds on a high-end consumer CPU. This performance was achieved by vectorizing the determination of background characteristics and source measurements using Numba's `guvectorize` decorator, alongside a large number of other optimizations with smaller impact.

Measured source parameters are now collected into a Pandas `DataFrame` making storage in HDF5 format trivial. Installation via `pip` is straightforward, and source extraction can be performed both standalone, or inside data-processing pipelines across multiple wavelength domains.

1. Introduction

PySE (Python Source Extractor) is the source finder for the Transients Key Project (Fender et al. 2006; Stappers et al. 2009)) for LOFAR (van Haarlem et al. 2013), that was mainly developed over the years 2005-2013 as part of the LOFAR Transients Pipeline (TraP, see Swinbank et al. (2015)). Figure 1 depicts TraP's workflow and the central role of PySE within it.

The source extraction and measurement workflow and its algorithms were heavily inspired by SExtractor (Bertin & Arnouts 1996), see Spreeuw (2010, chapter 2) for background on PySE's design. Spreeuw (2010, chapter 3) and Carbone et al. (2018) discuss its validation and results from expanded tests, respectively, while Spreeuw et al. (2018) contains a link to the source code. Despite many similarities between the two packages, there are also differences, which are discussed in Spreeuw (2010, chapter 2).

Shortly after the commissioning of the LOFAR telescope, the AARTFAAC (Wijers 2011; Prasad & Wijnholds 2012; Rowlinson 2012) project was initiated to find bright radio transients by monitoring the entire visible sky continously. The AARTFAAC pipeline (Prasad et al. 2016) used TraP as a subpipeline.

The goal of the present work is to make PySE sufficiently efficient to keep pace with the rate at which images are produced. In particular, PySE must be able to process very large images containing a large number of sources, as expected for SKA, the Square Kilometre Array (Lazio & Gaensler 2007). To this end, we use an artificial benchmark image of 4096×4096 pixels containing 167,281 sources, representative of images that may be generated every second.

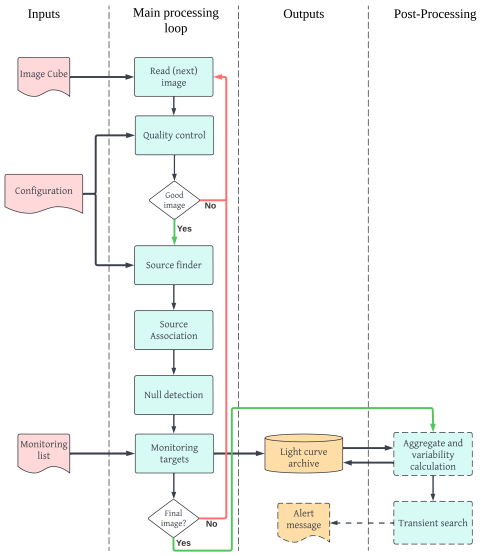


Figure 1. PySE is "Source finder" within The LOFAR Transients Pipeline (TraP). This workflow diagram was copied from Millenaar.

2. Methods

This section describes the two main performance optimizations applied to PySE's algorithmic components, both of which involve the Numba (Lam et al. 2015) Python compiler. Over the last years numerous other enhancements, each with less impact, have been implemented, but these are too specific to discuss here.

2.1. Vectorized $\kappa \cdot \sigma$ clipping over a grid of subimages

In order to calculate the mean and standard deviation of the background noise, one must separate the background pixels from the source pixels, through a technique called $\kappa \cdot \sigma$ clipping, see Bertin & Arnouts (1996, paragraph 2) for an explanation of this method. It is computationally expensive because the median and standard deviation (denoted by σ) have to be recomputed each time pixels outside the median $\pm\kappa \cdot \sigma$ range are clipped. Out of these operations, calculating the median is the most expensive because it involves sorting.

Sky temperature and telescope properties often result in varying background characteristics across an image. To account for this, the image is divided into into rectangular subimages on a regular grid. In PySE, $\kappa \cdot \sigma$ clipping is implemented as a function that works on a single subimage. By applying Numba's `guvectorize` decorator, pixel values across all subimages can be clipped in a vectorized way, yielding a considerable speedup. Here, vectorization refers to calculations applied to entire 3D NumPy `ndarray`s containing many flattened subimages rather than a single flattened subimage. The two additional dimensions correspond to the background grid, its nodes are centered on the subimages. This approach exploits SIMD (Single Instruction, Multiple Data) instructions at the CPU level within a single core. A decorator can be thought of as a function wrapper that adds functionality without modifying internal code. The `guvectorize` decorator not only allows for vectorization, but also for parallellization over multiple CPU cores: this resulted in an extra speedup.

2.2. Vectorize measurements over all sources

Measuring each source entails a large number of computations, which—if performed sequentially—would be problematic. For a 4096×4096-pixels benchmark image with 167,281 inserted sources, such an approach would not complete within one second. In order to solve this we chose the same approach as for deriving the background characteristics, i.e. we first wrote a function that computes all desired parameters of a single flattened island of pixel values, i.e., a single source represented as a 1D `ndarray`. Next, we applied Numba's `guvectorize` decorator to this function such that we could use a 2D `ndarray`, with each row correponding to a flattened island of pixel values, as input. It is important to note that this vectorization can only succeed if the "single-island function" makes use of Python and Numpy routines only, e.g. we cannot call Scipy routines. For this reason we do not fit Gaussians to sources, but apply "tweaked moments", see Spreeuw (2010, equations 2.66 and 2.67). The subsequent steps, which involve conversion to celestial coordinates, have also been vectorized across all sources using Astropy routines such as `wcs.WCS.all_pix2world` (Astropy Collaboration et al. 2022).

3. Results

3.1. Improved performance

To assess the performance of the accelerated source-finding pipeline, we applied the code to a synthetic benchmark image of 4096×4096 pixels containing 167,281 sources placed on a regular grid. In the intended observational setting, the radio telescope produces one image per second, implying that the full processing of a single image must be completed within this time budget.

All benchmarks were performed on a single PC equipped with an AMD Ryzen 9 7900X CPU (base clock 4.7 GHz), featuring 12 physical cores and 24 logical cores, resulting in an average of 812 ms for the best of five repetitions of the timing loop of five PySE runs. The reported runtime does not include the time required to load the image from disk. For images of this size, disk I/O can add approximately 300–400 ms. In a real-time transient pipeline, however, images are typically passed directly in memory.

Finally, we note that for images with very high source densities, estimating the background noise standard deviation accurately becomes increasingly challenging. To mitigate this, PySE allows the user to provide an externally generated background noise

map containing an interpolated estimate of the local standard deviation. When such a map is supplied, the runtime is reduced by approximately 150 ms, further increasing the margin with respect to the real-time processing requirement.

3.2. Obtaining a Pandas `DataFrame`

PySE exposes its results directly as a Pandas (The Pandas development team 2026; Wes McKinney 2010) `DataFrame` containing the measured parameters and associated uncertainties for all detected sources. A typical interactive usage pattern in an IPython environment, returning `source_parameters_df` as a Pandas `DataFrame`, is illustrated below:

```
from pathlib import Path
from sourcefinder.accessors.fitsimage import FitsImage
from sourcefinder.accessors import sourcefinder_image_from_accessor
from sourcefinder.config import Conf, ImgConf, ExportSettings

source_params = ["ra", "dec", "ra_err", "dec_err", "peak",
"peak_err"]

conf = Conf(
image=ImgConf(detection_thr=12.0, analysis_thr=8.0, grid=128,
vectorized=True),
export=ExportSettings(source_params=source_params),
)

fits_img = FitsImage(Path("new_observation.FITS"))
img = sourcefinder_image_from_accessor(fits_img, conf=conf)
source_parameters_df = img.extract()
```

The `vectorized=True` `ImgConf` attribute setting, which controls the source measurements, is redundant, since it is set to `True` by default. This implies that PySE's fast "sky-eater" mode is standard. It includes the use of "tweaked moments" for source measurements (see Spreeuw (2010, equations 2.66 and 2.67)). PySE reverts to Gaussian fits only when `vectorized=False` and/or deblending is turned on, i.e. when `deblend_nthresh>0`. In these cases, PySE's "sky-eater" mode is off.

For non-interactive use cases, PySE, which is pip-installable, provides a command-line interface, that comes with functions for visualisation, self-calibration, storage and display of source parameter values.

4. Conclusions

We have demonstrated that PySE meets the real-time performance requirements in the presence of very high numbers of sources. Beyond raw performance, PySE allows easy integration into transient detection pipelines by providing the measured source parameters as a Pandas `DataFrame`. Together, these features make PySE a practical and flexible tool for source extraction in next-generation astronomy surveys across multiple wavelength domains.

References

Astropy Collaboration, Price-Whelan, A. M., Lim, P. L., Earl, N., Starkman, N., Bradley, L., Shupe, D. L., Patil, A. A., Corrales, L., Brasseur, C. E., Nöthe, M., Donath, A., Tollerud, E., Morris, B. M., Ginsburg, A., Vaher, E., Weaver, B. A., Tocknell, J., Jamieson, W., van Kerkwijk, M. H., Robitaille, T. P., Merry, B., Bachetti, M., Günther, H. M., Aldcroft, T. L., Alvarado-Montes, J. A., Archibald, A. M., Bódi, A., Bapat, S., Barentsen, G., Bazán, J., Biswas, M., Boquien, M., Burke, D. J., Cara, D., Cara, M., Conroy, K. E., Conseil, S., Craig, M. W., Cross, R. M., Cruz, K. L., D'Eugenio, F., Dencheva, N., Devillepoix, H. A. R., Dietrich, J. P., Eigenbrot, A. D., Erben, T., Ferreira, L., Foreman-Mackey, D., Fox, R., Freij, N., Garg, S., Geda, R., Glattly, L., Gondhalekar, Y., Gordon, K. D., Grant, D., Greenfield, P., Groener, A. M., Guest, S., Gurovich, S., Handberg, R., Hart, A., Hatfield-Dodds, Z., Homeier, D., Hosseinzadeh, G., Jenness, T., Jones, C. K., Joseph, P., Kalmbach, J. B., Karamehmetoglu, E., Kałuszyński, M., Kelley, M. S. P., Kern, N., Kerzendorf, W. E., Koch, E. W., Kulumani, S., Lee, A., Ly, C., Ma, Z., MacBride, C., Maljaars, J. M., Muna, D., Murphy, N. A., Norman, H., O'Steen, R., Oman, K. A., Pacifici, C., Pascual, S., Pascual-Granado, J., Patil, R. R., Perren, G. I., Pickering, T. E., Rastogi, T., Roulston, B. R., Ryan, D. F., Rykoff, E. S., Sabater, J., Sakurikar, P., Salgado, J., Sanghi, A., Saunders, N., Savchenko, V., Schwardt, L., Seifert-Eckert, M., Shih, A. Y., Jain, A. S., Shukla, G., Sick, J., Simpson, C., Singanamalla, S., Singer, L. P., Singhal, J., Sinha, M., Sipőcz, B. M., Spitler, L. R., Stansby, D., Streicher, O., Šumak, J., Swinbank, J. D., Taranu, D. S., Tewary, N., Tremblay, G. R., de Val-Borro, M., Van Kooten, S. J., Vasović, Z., Verma, S., de Miranda Cardoso, J. V., Williams, P. K. G., Wilson, T. J., Winkel, B., Wood-Vasey, W. M., Xue, R., Yoachim, P., Zhang, C., Zonca, A., & Astropy Project Contributors 2022, ApJ, 935, 167. eprint: 2206.14220

Bertin, E., & Arnouts, S. 1996, A&AS, 117, 393

Carbone, D., Garsden, H., Spreeuw, H., Swinbank, J. D., van der Horst, A. J., Rowlinson, A., Broderick, J. W., Rol, E., Law, C., Molenaar, G., & Wijers, R. A. M. J. 2018, Astronomy and Computing, 23, 92. eprint: 1802.09604

Fender, R. P., Wijers, R. A. M. J., Stappers, B., Braun, R., Wise, M., Coenen, T., Falcke, H., Griessmeier, J.-M., Van Haarlem, M., Jonker, P. G., Law, C., Markoff, S., Masters, J., Miller-Jones, J. C. A., Osten, R., Scheers, B., Spreeuw, H., Swinbank, J., Vogt, C., Wijnands, R., & Zarka, P. 2006, in VI Microquasar Workshop: Microquasars and Beyond, 104.1. eprint: astro-ph/0611298

Lam, S. K., Pitrou, A., & Seibert, S. 2015, in Proceedings of the Second Workshop on the LLVM Compiler Infrastructure in HPC (New York, NY, USA: Association for Computing Machinery), LLVM '15. URL https://doi.org/10.1145/2833157.2833162

Lazio, T. J. W., & Gaensler, B. M. 2007, Highlights of Astronomy, 14, 390

Millenaar, T.in ADASS XXXV, edited by K. Polsterer, & S. Wagner (San Francisco: ASP), vol. TBD of ASP Conf. Ser., 999 TBD

Prasad, P., Huizinga, F., Kooistra, E., van der Schuur, D., Gunst, A., Romein, J., Kuiack, M., Molenaar, G., Rowlinson, A., Swinbank, J. D., & Wijers, R. A. M. J. 2016, Journal of Astronomical Instrumentation, 5, 1641008. eprint: 1609.04205

Prasad, P., & Wijnholds, S. J. 2012, arXiv e-prints, arXiv:1205.3056. eprint: 1205.3056

Rowlinson, A. 2012, in Gamma-Ray Bursts 2012 Conference (GRB 2012), 120

Spreeuw, H., Swinbank, J., Molenaar, G., Staley, T., Rol, E., Sanders, J., Scheers, B., & Kuiack, M. 2018, PySE: Python Source Extractor for radio astronomical images, Astrophysics Source Code Library, record ascl:1805.026. eprint: 1805.026

Spreeuw, J. N. 2010, Ph.D. thesis, University of Amsterdam, Netherlands. URL https://hdl.handle.net/11245/1.324881

Stappers, B., Fender, R. P., & Wijers, R. A. M. J. 2009, in 8th International e-VLBI Workshop, 21

Swinbank, J. D., Staley, T. D., Molenaar, G. J., Rol, E., Rowlinson, A., Scheers, B., Spreeuw, H., Bell, M. E., Broderick, J. W., Carbone, D., Garsden, H., van der Horst, A. J., Law,

C. J., Wise, M., Breton, R. P., Cendes, Y., Corbel, S., Eislöffel, J., Falcke, H., Fender, R., Grießmeier, J.-M., Hessels, J. W. T., Stappers, B. W., Stewart, A. J., Wijers, R. A. M. J., Wijnands, R., & Zarka, P. 2015, Astronomy and Computing, 11, 25. eprint: 1503.01526

The Pandas development team 2026, pandas-dev/pandas: Pandas. URL `https://doi.org/10.5281/zenodo.18328522`

van Haarlem, M. P., Wise, M. W., Gunst, A. W., Heald, G., McKean, J. P., Hessels, J. W. T., de Bruyn, A. G., Nijboer, R., Swinbank, J., Fallows, R., Brentjens, M., Nelles, A., Beck, R., Falcke, H., Fender, R., Hörandel, J., Koopmans, L. V. E., Mann, G., Miley, G., Röttgering, H., Stappers, B. W., Wijers, R. A. M. J., Zaroubi, S., van den Akker, M., Alexov, A., Anderson, J., Anderson, K., van Ardenne, A., Arts, M., Asgekar, A., Avruch, I. M., Batejat, F., Bähren, L., Bell, M. E., Bell, M. R., van Bemmel, I., Bennema, P., Bentum, M. J., Bernardi, G., Best, P., Bîrzan, L., Bonafede, A., Boonstra, A.-J., Braun, R., Bregman, J., Breitling, F., van de Brink, R. H., Broderick, J., Broekema, P. C., Brouw, W. N., Brüggen, M., Butcher, H. R., van Cappellen, W., Ciardi, B., Coenen, T., Conway, J., Coolen, A., Corstanje, A., Damstra, S., Davies, O., Deller, A. T., Dettmar, R.-J., van Diepen, G., Dijkstra, K., Donker, P., Doorduin, A., Dromer, J., Drost, M., van Duin, A., Eislöffel, J., van Enst, J., Ferrari, C., Frieswijk, W., Gankema, H., Garrett, M. A., de Gasperin, F., Gerbers, M., de Geus, E., Grießmeier, J.-M., Grit, T., Gruppen, P., Hamaker, J. P., Hassall, T., Hoeft, M., Holties, H. A., Horneffer, A., van der Horst, A., van Houwelingen, A., Huijgen, A., Iacobelli, M., Intema, H., Jackson, N., Jelic, V., de Jong, A., Juette, E., Kant, D., Karastergiou, A., Koers, A., Kollen, H., Kondratiev, V. I., Kooistra, E., Koopman, Y., Koster, A., Kuniyoshi, M., Kramer, M., Kuper, G., Lambropoulos, P., Law, C., van Leeuwen, J., Lemaitre, J., Loose, M., Maat, P., Macario, G., Markoff, S., Masters, J., McFadden, R. A., McKay-Bukowski, D., Meijering, H., Meulman, H., Mevius, M., Middelberg, E., Millenaar, R., Miller-Jones, J. C. A., Mohan, R. N., Mol, J. D., Morawietz, J., Morganti, R., Mulcahy, D. D., Mulder, E., Munk, H., Nieuwenhuis, L., van Nieuwpoort, R., Noordam, J. E., Norden, M., Noutsos, A., Offringa, A. R., Olofsson, H., Omar, A., Orrú, E., Overeem, R., Paas, H., Pandey-Pommier, M., Pandey, V. N., Pizzo, R., Polatidis, A., Rafferty, D., Rawlings, S., Reich, W., de Reijer, J.-P., Reitsma, J., Renting, G. A., Riemers, P., Rol, E., Romein, J. W., Roosjen, J., Ruiter, M., Scaife, A., van der Schaaf, K., Scheers, B., Schellart, P., Schoenmakers, A., Schoonderbeek, G., Serylak, M., Shulevski, A., Sluman, J., Smirnov, O., Sobey, C., Spreeuw, H., Steinmetz, M., Sterks, C. G. M., Stiepel, H.-J., Stuurwold, K., Tagger, M., Tang, Y., Tasse, C., Thomas, I., Thoudam, S., Toribio, M. C., van der Tol, B., Usov, O., van Veelen, M., van der Veen, A.-J., ter Veen, S., Verbiest, J. P. W., Vermeulen, R., Vermaas, N., Vocks, C., Vogt, C., de Vos, M., van der Wal, E., van Weeren, R., Weggemans, H., Weltevrede, P., White, S., Wijnholds, S. J., Wilhelmsson, T., Wucknitz, O., Yatawatta, S., Zarka, P., & Zensus, A. 2013, A&A, 556, A2. eprint: 1305.3550

Wes McKinney 2010, in Proceedings of the 9th Python in Science Conference, edited by Stéfan van der Walt, & Jarrod Millman, 56

Wijers, R. A. M. J. 2011, in American Astronomical Society Meeting Abstracts #217, vol. 217 of American Astronomical Society Meeting Abstracts, 223.02

Astronomical Data Analysis Software and Systems XXXV
*ASP Conference Series, Vol. **VOL** Compiled: Tuesday 30th June, 2026@ 01:30*
K. Polsterer, D. Bohlender, S. Gwyn, A. Hincks and P. Teuben, eds.
*© **YEAR** Astronomical Society of the Pacific*

ZZZDEBUG!055.tex
Wipper, C.|textbf
Usher, C.
observing!scheduling!queue

Kealahou: A Look at the Past, Present, and Future of CFHT's New Queued Service Observing System

C. Wipper and C. Usher

Canada-France-Hawaii Telescope, Kamuela, HI, USA;
wipper@cfht.hawaii.edu , usher@cfht.hawaii.edu

Abstract. After 45 years in operation, and two decades of remote queued service observing (QSO), Canada-France-Hawaii Telescope is developing Kealahou: a reconstruction of our entire QSO software system. Kealahou started as a platform for a single spectrograph, and has since grown to host CFHT's main suite of instruments, a "Phase 1" proposal submission and review system, and science operations management tools. With these features, the complete QSO process, from proposal creation to data access, is now encompassed by a unified API and web application. Here we describe the capabilities of the system, as well as the pros and cons of the development approach, and a look towards the future.

1. Kealahou: "The New Way"

Canada-France-Hawaii Telescope (CFHT) has been operational for over 45 years. In that time, observational astronomy has gone through many evolutions made possible by the continued application of new technologies and methodologies, from the replacement of photographic plates with CCDs to the transition to service observing and remote operations. CFHT has been a leader in these advancements, allowing it to remain at the forefront of astronomical discovery despite the construction of progressively larger and more powerful telescopes. One of the observatory's latest evolutions is Kealahou, a reconstruction of our entire queued service observing (QSO) software system.

2. Why redesign the QSO software system?

The decision to redesign and rebuild CFHT's QSO software system was prompted by five key factors:

1. Over the years, as new instruments and larger science programs were added to CFHT's operations, the QSO database needed increasingly complex data models, while taking in an ever-growing volume of data. This put significantly more strain on the database, causing technical issues during science operations as it failed to keep with demands.

2. Tools had serious usability problems. User accounts and science programs had to be manually created by technical staff at the start of each semester. Archaic UI design confused PIs who were filling out their program details. PIs had share

tals!CDS
abases!individual!Simbad
alogues!services!VizieR
lications!Aladin Lite

their password with collaborators. Idle sessions caused users to be locked out, requiring an admin override. Uploaded files would sometimes vanish.

3. Many of the technologies used in the QSO software stack were old, outdated, and proprietary. This posed security concerns, and some components would stop working as users updated their web browsers. In some cases, these tools had no clear upgrade path. For others, the path did exist but was clearly leading to an eventual dead-end.

4. Software systems that were related conceptually were totally disconnected in practice, with no way to share data or communicate. For example, the Phase 1 proposal system for was completely separate from the Phase 2 system for programs, requiring both CFHT staff and PIs to manually track and re-enter data.

5. In the modern era of astronomy with the widespread adoption of Python, many users are more code-savvy, leading to a wider demand for the ability to programmatically manage programs with more complex observing strategies. Looking forward, programmatic access was also of internal interest to allow automation.

3. What Kealahou does

Kealahou currently encompasses the following capabilities and features, which together form CFHT's end-to-end QSO system:

- A user management and authentication system, with self-service sign up and account management tools.
- K1, a structured 'Phase 1' science proposal submission interface for PIs, supporting concurrent calls and multi-instrument proposals.
- A full proposal review system for TACs, including review management and reviewer assignments, and proposal feedback to PIs.
- Internal proposal and program management tools for observatory staff, with consolidated overviews as well as proposal/program-specific interfaces.
- K2, a full 'Phase 2' program management interface for science observation request submission, featuring CDS target lookup and automated finding charts.
- A self-serve interface for PIs to manage their collaborators in both K1 and K2.
- A scheduling interface for querying observations, preparing nightly observation queues, and grading and validating completed observations.
- A web-based graphical observation execution interface used by service observers.
- Execution of instrument and telescope commands for all configurable observing modes, across all instruments, including SNR goal-based observations.
- Data acquisition notifications for new program data and direct access to reduced science data through a web interface and command-line tools.

- Automation for observatory processes, such as instrument calibration sequences, time accounting calculations, and program and proposal statistics.
- Complete user documentation for K1 and K2.
- Public access to nightly observation reports and weather logs.

databases!Sybase
databases!MariaDB
libraries!Angular
computer languages!JavaScript
web!browser
techniques!API
protocols!HTTP
data formats!JSON
computer languages!Java

4. Building a custom platform for the modern era

Kealahou replaces our use of obsolete, proprietary database and web platforms with modern, open-source technologies. System logic has been extracted from the UI and database into an API-based application layer, providing new flexibility to create tools that interface with the system. The software is built and deployed through an automated pipeline.

The backbone of the Kealahou is its database. We replaced the Database Management System (DBMS), moving from the proprietary Sybase ASE 12.5, released in 2001 and EOL since 2009, to the open-source and actively developed MariaDB. We also revamped the data models to improve performance and give the system more flexibility going forward. One specific change we made was to use data "blob" fields for each entity rather than storing all data directly in columns.

The top layer of the Kealahou system is the web interface. This unified web application provides users a single UI to access all stages of the QSO process. The existing tools replaced by this new interface were using a slew of server-side and client-side technologies, including ColdFusion 5 from 2001. The new application is built in TypeScript and Angular, and served as just HTML, CSS, and JavaScript, allowing it to work in modern browsers.

At the core of Kealahou lies a web API that the web interface and other client applications use to access the database and to trigger actions for observatory operations. The API, created with Protocol Buffers, is accessible through REST-like HTTP endpoints using JSON data, as well as through gRPC methods for internal use.

Unlike the previous QSO system, which used database stored procedures for system logic, in Kealahou this is all done in the API server application, written in Java. Tools fetching data through the Kealahou API receive complete entities without needing to manually join rows in database queries. Additionally, post-processing previously done by separate programs and scripts is now directly baked in.

5. Development strategy pros and cons

5.1. Old and new, operating in parallel

When development began, it was decided that this new system would run in parallel with CFHT's legacy QSO applications and databases. Data would be shared between the systems by having Kealahou write updates back to the legacy database, and by regularly running a database "syncer" task to copy data from the legacy database into Kealahou. Initially, Kealahou would be developed to run SPIRou, CFHT's latest incoming (at the time) instrument, and existing instruments would later be ported over one by one.

get of opportunity
orithm!automatic

This reduced the risk of seriously disrupting operations with a bug or issue in the new system. This approach also allowed for incremental improvements and gradually decreased strain on the old system as each instrument was migrated.

However, having two systems split by instruments caused confusion for PIs (and even staff). Over time, the development effort spent on ensuring compatibility scaled not just with the size of glue code, but also test code, and to some extent the entire code base. Finally, we underestimated the performance impact when syncing between systems, particularly the spikes during periods of heavy usage.

5.2. A broadly scoped software project

The Kealahou project was scoped to encompass a wide array of software responsibilities for the observatory, including functionality for both internal and external users. While not strictly necessary to fulfill these requirements, this naturally led the software components of Kealahou to have a similarly monolithic design.

This allowed us to quickly develop a broad set of features for rapid iteration early in the project's life. It also reduced the work needed to have systems which were not closely related interact with each other, and to maintain functional compatibility between subsystems. Overall, it was easier to create a cohesive user experience.

On the other hand, some work done early on to support broad requirements ended up being at odds with further development as requirements shifted over the life of the project. The monolithic nature of the code base meant that it was difficult to on-board developers to work on a subset of the project, and that there was more effort to maintain unrelated subsystem code when refactoring. This design also made it impossible to take down just one subsystem for maintenance.

Overall, when taking such incremental or expansive approaches to development, it is important to remain cognizant of the challenges doing so may cause, and to evaluate how (and when) to best mitigate the negative effects while keeping the positives.

6. Kealahou in the age of LSST and AI

Starting in 2026, CFHT users will be able to submit targets and observation requests to Kealahou from the AEON network using the Kealahou API. AEON integration will support both fixed and moving targets. Additionally, this project will implement of a rapid Target of Opportunity (rToO) capability, cutting ToO observation delays from 'hours-to-days' to 'seconds-to-minutes'. As part of this project, we will publish a full API specification, opening up new opportunities for users to interact with CFHT in a programmatic way, either directly, or through their own software tools and systems.

Following this, we have some long-term ideas for Kealahou, including automatic queue scheduling, autonomous observing, integration of natural language processing, and spinning off K1 or other parts of Kealahou into open-source modules.

With the application now responsible for the full end-to-end QSO operations at CFHT, and a long list of future developments, Kealahou is expected to serve CFHT well into the next decade.

Acknowledgments. The Kealahou Team would like to acknowledge CFHT's privileged location on Maunakea, a significant historic and cultural site to the indigenous Hawaiian people. The Kealahou Team conducts all development with the utmost care and respect for Maunakea.

Astronomical Data Analysis Software and Systems XXXV
*ASP Conference Series, Vol. **VOL** Compiled: Tuesday 30th June, 2026@ 01:30*
K. Polsterer, D. Bohlender, S. Gwyn, A. Hincks and P. Teuben, eds.
*© **YEAR** Astronomical Society of the Pacific*

ZZZDEBUG!057.tex
Shortridge, K.|textbf
visualisation
data formats!FITS
data
formats!FITS!headers
data
formats!FITS!keywords
computer
languages!Python
FOOBAR! ascl:1101.010

Data Formats and Visualisation BoF

Keith Shortridge

K&V, Sydney, NSW, Australia; KeithShortridge@gmail.com

Abstract. ADASS used to hold a regular FITS BoF (Birds of a Feather meeting). As other data formats started to be used along with FITS, this became a Data Formats BoF, and there was some element of competition between formats, together with an occasional attempt to create a unified Format that could replace FITS as the only astronomical format needed. The impetus for this year's BoF came from an acceptance that astronomy would have to work with multiple formats in the foreseeable future, and a question: Did the use of multiple formats make it difficult to write general-purpose utilities, for example visualisation programs such as SAOImage/DS9, and if so was this a problem? The resulting discussion was interesting, and although it came to no ultimate conclusion, it at least made it clearer why such a conclusion was hard to reach.

1. Introduction, and a little history

In introductory remarks, the issue was framed as follows:

Some people can remember the days before FITS, the wild frontier days when every postdoc had their own data format and their own program to read it. FITS changed all that, and there was a golden age when most astronomical data (and some from the Vatican, which FITS was very proud of) were written in FITS, and, because the format was settled, people could write general-purpose programs like SAOImage that could display almost everyone's images.

But FITS was a 'transport' format., and it wasn't easy to manipulate FITS files on disk. So people writing data reduction programs came up with other formats whose data could be modified in place. And some of these implemented a hierarchical format, which provided a lot of flexibility. But they weren't designed for data interchange, and none of them were a standard, so people kept using FITS for that. But FITS couldn't easily handle everything you could put in a hierarchical format, and perhaps that's where the drift from FITS started. And this accelerated with the introduction of formats like HDF. By the time HDF got to HDF5 it had settled down into a de facto standard of sorts. And it's used a lot. In many different ways. But not everyone was comfortable with HDF5, and it's not the only game in town – ADASS 2024 included an ASDF tutorial, for example.

And the FITS BoFs (e.g. Mink et al. (2019) became data formats BoFs (e.g. Mink et al. (2020)), and people tried – and failed – to come up with a new all-encompassing standard. And even if you fix on something really flexible like HDF5, there are so many ways of using it, so many schemas, that just saying this is in such and such a hierarchical format doesn't mean you know where to find the main data array – if there is one – and you certainly may not know where to look for its coordinate data.

It looks as if, in the near future, we aren't going to be all agreeing on a universal data standard. So surely that makes it pretty hard to write an SAOImage that can handle any astronomical data file you throw at it.

Three questions:

- Is it the case that it's now difficult to write general purpose programs that can handle any astronomical data?
- If so, is this a problem? (And let's remember that we may not be a representative sample. Most people here could throw together a purpose-built astropy program to read a given data file and do something with the data in it. But can every astronomer?)
- If it is a problem, can we do something about it?

As might have been expected, the discussion didn't quite follow that neat and logical sequence.

2. Discussion

At first, it seemed there was a consensus that it was now hard to write general-purpose programs able to handle a variety of data files. However, as discussion danced around this topic, an alternative view emerged, led mostly by comments from Erik Tollerud, to the effect that this was not a fundamental problem; all that was really required was to split the problem into layers, and get the data into your system via the appropriate layer.

Although such a description isn't specific to a Python-based system, it was clearly a point of view influenced by, say, an astropy approach, where it is easy to produce a new program by picking and choosing from a large selection of different modules, each with their own capabilities. Certainly, astropy has modules that can read FITS files and modules that can read HDF5 and other formats, and it's possible to mix and match these to produce programs that can take a variety of disparate data formats as input.

Does this solve the problem in principle? And does it solve it in practice?

Cleary, this sort of approach works best if there are modules that can read all the individual flavours of data file, which begs the question of who provides these modules. And using these modules becomes easier if they all conform to a standard interface that can be used by the higher levels. That may also make them easier to implement, because the writers don't have to decide for themselves what the higher levels would find useful. But that begs the question of how the standard interface gets defined.

The question of defining an interface got a little sidetracked by a question of language. Erik Tollerud pointed out that an interface is easier to specify in a given language, and there was a digression prompted by the claim that in 40 years there would still be a way to run Python. This may well be true – there are emulators now for some very old games consoles – and a clear definition remains a clear definition, independent of language, but that may not be the point.

A definition of interfaces moved almost inevitably to a discussion of data models; you need a clear picture of what your data contains in order to define an interface to it. This is also connected with the distinction between syntax and semantics – syntax is

a matter of actual layout, that a file can contain arrays, and structures, and values and how these are accessed, while semantics is how they are to be interpreted; which array is the main data array, which keywords define the coordinate transformation, etc.

With people in the room who had been involved in previous attempts to define an overarching model for astronomical data (and the subsequent understandable failure to converge to any final result) it was acknowledged that searching for an all-encompassing model was letting the perfect be the enemy of the good, and discussion moved to whether it would be possible to define a model of a useful subset of data.

Mixed in with this was discussion of who would write the access modules that sat between a given data format and the higher levels, the modules that would read that format and make it available using the (still TBD) standard interface. There seemed to be a consensus that these would best be provided by the people responsible for generating the data, and also a feeling that not all those responsible would do so.

Examples were provided contrasting formats using HDF5 (which essentially defines the syntax) that provided clear schemas explaining the function of the various components (ie the semantics) with others that lacked such clear schemas. However, it was also remarked that locking oneself into a fixed schema, a strict scheme with a tool that enforces it, could get in the way at a later date, and this was perhaps a fundamental problem.

Was it possible to define an abstract data model that described a useful subset of astronomical data? Towards the end of the discussion it was suggested that the community did have a concrete example of a subset data model that was undeniably useful. And this was FITS. Was this a possible, if limited, solution?

Define an abstract data model that provides access to data arrays, a set of scalar values, and even tables of values, as FITS does, and define an interface – an API – to this, implement it for a number of formats, and then you could create a program that could work with any of those formats. It might not make use of everything that format could contain, but you could, for example, display the images, ideally with coordinate information. But then someone burst the bubble by saying; 'well, you could just convert the file to FITS and read that in'. Since most people creating non-FITS files do provide at least a minimal FITS conversion program, introducing a complicated software layer that merely did that conversion in situ seems redundant.

Since time had also run out, that seemed a natural place to stop the discussion.

At least for the moment.

3. Reflections

In the end, did this BoF realise that FITS provided a useful minimal concrete data model, but at the same time also demonstrate that creating a program layer and API implementing such a model was unnecessary, because one could always just convert a non-FITS file to FITS anyway?

Arguably, yes. But is there a loss of flexibility here? Perhaps an early question illustrates this. The discussion had considered a program that used a module targeted at a given non-FITS format that implemented a defined interface, which would allow the program to access files in that format. The question was: could this program also read FITS? The simple answer was no. However, the same program could be linked with

an equivalent module that implemented the same interface but for FITS files. Having that defined interface meant that, if both modules implementing that interface were available, producing both versions of the program would be trivial, even in a language like C++. In Python it might need no more than another import line.

And that actually sounds useful. It would be even more useful if a single program could be linked with modules for a number of different formats, and yet more useful if the interface also allowed for access to some of those features of hierarchical formats that have drawn people to those formats.

At a minimum one could imagine that the interface might simplify the view of the data. If it defined a call that provided access to the main data in the file as an N-dimensional array and another that returned the N coordinates for an element of that array, no matter how stored in the format, this would simplify a lot of use cases. Perhaps it could include a call to access the observation details such as exposure time in a standard form such as that defined by ObsCore. Barloy & Landais (2026) describe the use of mapping files to get such information from FITS files that follow their own rules as to keyword usage, and that concept could be extended to other formats.

It might help to have a standardised way of describing elements of hierarchical files that could be used in such mapping files to indicate the location of such values. This might allow a general module that understood HDF5 syntax in general to use a mapping file to help understand the semantics of a particular schema. (SAOImage has something of this functionality for handling some additional formats, but these do not yet support HDF5.)

The warning about allowing the perfect to be the enemy of the good applies, and it's too easy to get carried away. But maybe a limited interface that was just a little simpler than the way FITS files are accessed and which provided just a little more functionality than FITS might be a useful thing to have? After all, most people in the discussion had seemed to think it sounded like a good idea at first.

Perhaps next year.

4. References

References

Barloy, N., & Landais, G. 2026, in ADASS XXXV, edited by TBD (San Francisco: ASP), vol. TBD of ASP Conf. Ser., TBD

Mink, J., Diaz, R., Fernique, P., Landais, G., Mireille, L., & Michel, L. 2020, in ADASS XXIX, edited by R. Pizzo, E. R. Deul, J. D. Mol, J. de Plaa, & H. Verkouter, vol. 527 of ASP Conf. Ser., 761

Mink, J., Diaz, R., Shortridge, K., & Jenness, T. 2019, in ADASS XXVII, edited by P. J. Teuben, M. W. Pound, B. A. Thomas, & E. M. Warner, vol. 523 of ASP Conf. Ser., 701

Astronomical Data Analysis Software and Systems XXXV
*ASP Conference Series, Vol. **VOL** Compiled: Tuesday 30th June, 2026@ 01:30*
K. Polsterer, D. Bohlender, S. Gwyn, A. Hincks and P. Teuben, eds.
*© **YEAR** Astronomical Society of the Pacific*

ZZZDEBUG!058.tex
Shortridge, K.|textbf
libraries!OpenGL
software!tools!OpenGL
libraries!Vulkan
software!tools!Vulkan
visualisation!3D
software!tools!Qt
FOOBAR! ascl:1101.010

AAOGlimpse goes to Vulkan – and learns some lessons

Keith Shortridge

K&V, Sydney, NSW, Australia; KeithShortridge@gmail.com

Abstract. AAOGlimpse is an image display program originally based on Qt and OpenGL. In the interests of long-term portability and support, it is now being restructured to support the use of Vulkan for 3D graphics.

This paper describes some of the lessons learned in this process, perhaps the main one being that, in hindsight, the original version should not have been written quite the way it was.

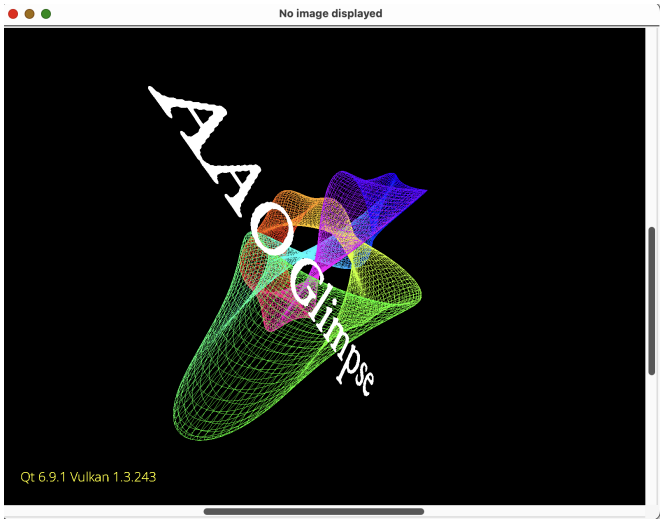


Figure 1. The AAOGlimpse startup screen. The figure rotates and the text moves around the screen with it.

1. Introduction

AAOGlimpse (Shortridge 2012) is a rather quirky, niche, image display program that displays a 2D image as a 3D surface and encourages the user to spin it around and play with it. It particularly appeals to instrument builders and detector engineers, who want to see any image defects clearly, and who find it convenient that it runs under Windows as well as under MacOS and Linux. Back in 2011, it was written using Qt for the user interface and an early version of OpenGL for 3D graphics. It was structured to allow moving to a GUI layer other than Qt, but use of OpenGL permeated the whole code. That version of OpenGL was deprecated in OpenGL 3.1, OpenGL development has

ceased, and OpenGL itself has been deprecated by Apple. AAOGlimpse is now being restructured to work through an abstract 3D graphics layer, the initial version of which uses Vulkan.

2. Program structure and working with Qt and Vulkan

Originally, two assumptions had been made in the design of AAOGlimpse, and at least one of these was wrong.

Firstly, that there needed to be a clearly-defined system-dependent layer containing all the Qt code, because Qt might need replacing at some point. (This might still prove to be correct.) Secondly, that there was no need to isolate the OpenGL code in one OpenGL-specific layer, as OpenGL was seen as a safe bet, being essentially the only platform-independent graphical sub-system. (In hindsight, this was clearly wrong.)

As a result, a lot of effort went into isolating the Qt code into one module. However, OpenGL calls were scattered enthusiastically throughout the code, which was easy at the time, and all that has now had to be taken apart and redone. (**Unless you're very lucky, your design decisions will come back to haunt you.**) (But: **Doing things properly is hard, and whether it's worth it in the long-run can be a trade-off.**)

Vulkan is a very low-level system. It emphasises flexibility over convenience, and it is often said that it takes 1000 lines of code to get it to draw a triangle. Those 1000 lines are complicated. (**Vulkan is not for the faint-hearted.**) But put a loop at the end, and that same code will draw a million triangles. Fortunately, most of those 1000 lines are boilerplate code that can be found in examples on the Web, particularly the excellent vulkan-tutorial.com (**The internet – and even AI, used cautiously – can be your friend.**)

The latest Qt versions provide a QVulkanWindow class that packages the use of Vulkan for graphics. However, using this might have bound the code too much to Qt. Also, a KVVulkanFramework class had already been developed – mostly to experiment with GPU computation, but also with graphics support – and it would be a useful learning exercise to extend it to support AAOGlimpse.

Fortunately, Qt provides a flexible interface that allows a conventional Qt window (a QWindow) to be used with Vulkan, and this can work nicely with a non-Qt Vulkan framework such as KVVulkanFramework.

Even so, supporting the things that AAOGlimpse needed, such as transparency, rotation in 3D, multiple graphics windows, and text in a graphics window (a notable can of worms) proved non-trivial. (**Sometimes simple things, like writing a string of text, can be surprisingly hard.**) An unexpected omission turned out to be support for drawing lines wider than one screen pixel – not supported on all platforms by Vulkan, and hard to emulate: a thick line in 3D is really a rod, and all you have to draw it with are triangles. (**Try to live without thick lines if you can.**)

The rotating Tranguloid Trefoil shape shown in figure 1 uses code from an Apple OpenGL demonstration program credited to Roger Bagula and Paul Bourke. Early OpenGL code had a global set of rotation matrices that systems like Vulkan move to individual shader modules coded to run on the GPU – giving flexibility at the expense of complexity. The text, which rotates with the trefoil figure, uses TrueType fonts rendered into a texture whose characters are then mapped onto appropriately positioned transparent triangles. The OpenGL version used the FTGL library for this, but with no obvious replacement for this under Vulkan, this had to be reimplemented from scratch. The

fixed position characters at bottom left are now drawn using a separate GPU pipeline running a different shader that does not apply a rotation matrix (simpler than manipulating the OpenGL global matrices to get the same effect).

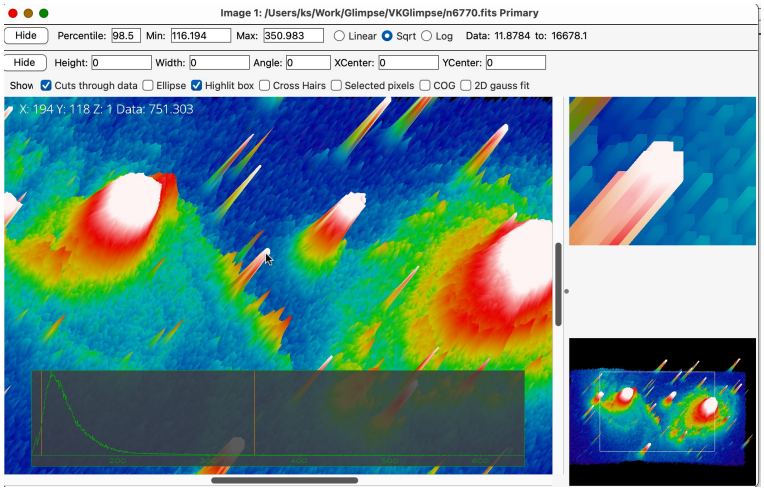


Figure 2. This shows an image of NGC6770, with the Zoom and Navigation windows enabled, and a superimposed plot of the data histogram with the current display scale limits, which can be dragged to accentuate features of the data.

The global nature of OpenGL caused problems with multiple windows (shown in figure 2). This was simpler in Vulkan, but required separate instances of the graphics sub-system for each window. Drawing the transparent line plot unrotated on top of the rotated display using a separate pipeline was also simpler in principle, but needed a deal of new code.

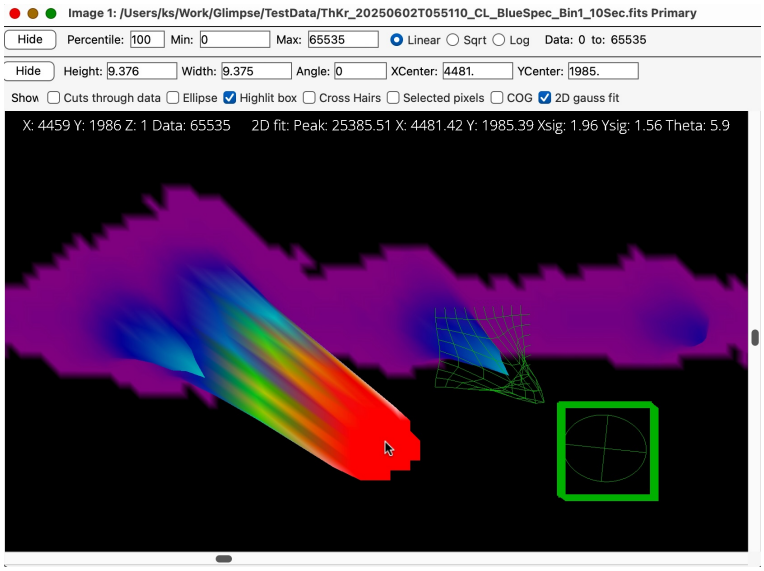


Figure 3. Working out which pixel is under the mouse was much easier in Vulkan.

Figure 3 is a close-up of arc lines in a calibration spectrum. The strongest line is saturated, and the saturated pixels are high-lit in red. Colouring pixels is easy. Fitting a 2D gaussian on the fly to the nearby unsaturated line is easy for the CPU. What's hard is working out which image pixel is being displayed under the mouse. If an image is displayed face-on, as it is in most 2D image displays, then it's trivial to work out which image pixel is at which screen position. But when the 2D image is turned into a 3D image and rotated, many image pixels may be projected onto any given screen position.

Ideally, you take every image pixel in turn, work out where it is projected on the screen, and the one that appears closest to the viewer is the one you want. Especially for large images, this is impractical for a CPU, and this was an unsolved problem for the original AAOGlimpse. But this is a massively parallel problem, and the GPU already knows the location of each basic triangle it used to generate the display. Now, as the mouse moves, the GPU constantly repeats the calculation for every image pixel, and here at top-left of the screen it shows the location and the saturated 65535 value of the image pixel under the mouse. (The latest OpenGL versions would also have supported this, but the early version used originally did not.)

3. Some other lessons learned

A stranger drives into a small village and asks a passing local how to get to some distant town. The local nods, thinks for a while, and finally says, "Well, if I were you, mate, I wouldn't start from here..."

- **3D rotation matrices are not for the mathematically-challenged.** (Two days were lost to a bug eventually traced to two matrices being multiplied together in the wrong order.)
- **AI-generated code can be instructive, but when it doesn't work it's hard to follow it well enough to fix it.** (A little AI code was used experimentally for the text handling. It almost worked.)
- **Trying something like this without knowing Vulkan well is rash.** (**But one way to get to know Vulkan well is to try something like this.**)

But mostly, **it's best to get the structure right the first time, rather than to have to fix it later.**

4. References

References

Shortridge, K. 2012, in ADASS XXI, edited by P. Ballester, D. Egret, & N. P. F. Lorente, vol. 461 of ASP Conf. Ser., 49

Astronomical Data Analysis Software and Systems XXXV
*ASP Conference Series, Vol. **VOL** Compiled: Tuesday 30th June, 2026@ 01:30*
K. Polsterer, D. Bohlender, S. Gwyn, A. Hincks and P. Teuben, eds.
*© **YEAR** Astronomical Society of the Pacific*

ZZZDEBUG!060.tex
Gómez, P.|textbf
Ruhberg, L. E.
Remmelgas, K., A.
O'Ryan, D.
FOOBAR!conference!ADASS 2019
FOOBAR!organisations!ASP
FOOBAR! ascl:1101.010

Cutana: A High-Performance Tool for Astronomical Image Cutout Generation at Petabyte Scale

Pablo Gómez,[1] Laslo E. Ruhberg,[2] Kristin A. Remmelgas[1], Jan Reerink[1] and David O'Ryan[1]

[1]*European Space Agency (ESA), European Space Astronomy Centre (ESAC), Camino Bajo del Castillo, Villanueva de la Cañada;* `pablo.gomez@esa.int`

[2]*Astronomisches Rechen-Institut (ARI), Zentrum für Astronomie, Universität Heidelberg, Mönchhofstr. 12-14, 69120 Heidelberg, Germany*

Abstract. The *Euclid* Quick Data Release 1 (Q1) with 30 million sources marks the start of the petabyte-scale data delivery of the telescope. Working on that scale requires extracting millions of source-specific cutouts, yet standard tools like `Astropy`'s Cutout2D process sources individually, creating bottlenecks for large catalogues. We introduce `Cutana`, a memory-efficient software tool optimised for batch processing in both local and cloud-native environments. `Cutana` employs vectorised `NumPy` operations to extract cutout batches simultaneously from FITS tiles, implements automated memory-aware scheduling, and supports both Zarr and FITS output formats with multiple common normalisation schemes (asinh, log, zscale). `Cutana` outperforms `Astropy` in all tested Q1 subset scenarios achieving near linear scaling and processing thousands of cutouts per second. On just four worker threads, Cutana can process all of Q1 in under four hours. The tool includes an `ipywidget` interface for parameter configuration and real-time monitoring. Integration with ESA Datalabs is underway for the *Euclid* DR1 release, with an open-source release available.

1. Introduction

Modern astronomical surveys are entering an unprecedented era of data generation. Telescopes like ESA's *Euclid* mission (Euclid Collaboration et al. 2025b) and the Vera C. Rubin Observatory (Ivezić et al. 2019) produce petabyte-scale datasets of billions of sources. *Euclid*'s Quick Data Release (Q1) provides 63.1 square degrees of observations of approximately 30 million objects (Euclid Collaboration et al. 2025a), foreshadowing the massive data volumes expected from the full survey. These datasets often harbour rare phenomena amenable to large-scale searches (O'Ryan & Gómez 2025).

Maximising scientific return from these datasets fundamentally relies on generating source-specific image cutouts for analysis. However, the infrastructure required to process such volumes presents significant challenges. Modern computational platforms like ESA Datalabs (Navarro et al. 2024) provide Kubernetes clusters and require following high-performance computing practices such as memory management.

The community's standard approach using `Astropy` processes sources individually through its Cutout2D implementation, creating substantial overhead when handling millions of objects. While `Astropy` provides robust world coordinate system handling and FITS manipulation, its single-source processing introduces bottlenecks. This leaves topics such as parallelisation and memory management to the user.

Hence, we present `Cutana`, a purpose-built tool designed to address these limitations through vectorised, load-balanced batch processing while maintaining compatibility with astronomical data standards and formats. The software combines efficient algorithms with dynamic resource management, making it suitable for deployment in

both traditional computing environments and modern containerised infrastructure. A streamlined graphical user interface (shown in Figure 1) enables rapid cutout creation.

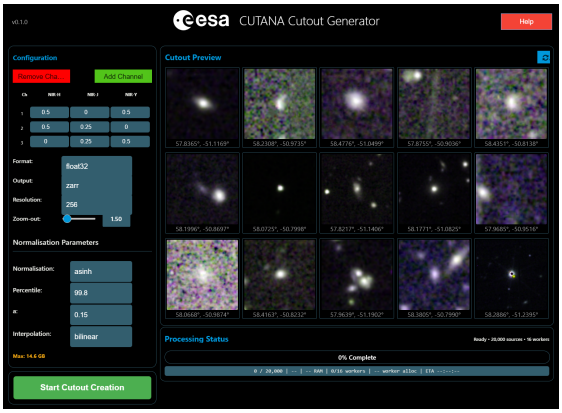


Figure 1. `Cutana`'s ipywidget-based interface within ESA Datalabs showing configuration panel (left) alongside real-time cutout preview grid (right).

2. Implementation

`Cutana` is implemented as a Python package with modular components for data processing, I/O operations, and user interaction. The architecture enables both interactive usage through Jupyter notebooks and automated execution in pipeline environments.

The backend employs a multi-stage pipeline optimised for astronomical image processing. The input source catalogues provide celestial coordinates (right ascension, declination), object sizes, and FITS file paths. `Cutana` groups sources by their parent tiles, enabling efficient batch extraction through vectorised `NumPy` operations. The pipeline stages include: (1) catalogue validation and source grouping by tile, (2) tile-based batch processing with memory-mapped FITS access, (3) coordinate transformation and cutout extraction using optimised array slicing, (4) normalisation and format conversion, and (5) metadata generation and output packaging. The system implements dynamic load balancing through continuous memory monitoring, adjusting worker processes based on available resources.

`Cutana` supports both traditional FITS and the cloud-optimised Zarr format (Miles et al. 2025). Zarr provides efficient storage for multidimensional arrays through chunked and compressed storage that enables parallel I/O operations. Output data includes comprehensive metadata in parquet format: source identifiers, coordinates, processing timestamps, and array indices, enabling rapid source retrieval from batch-processed archives. `Cutana` leverages `fitsbolt` for high-performance FITS manipulation and `images-to-zarr` for efficient format conversion.

The software incorporates multiple image normalisation methods essential for astronomical analysis, powered by the `fitsbolt` library. Available stretches include asinh, log, and zscale transformations, each with configurable parameters unified under a consistent interface. The system supports channel blending for multi-band observations, enabling users to combine multiple FITS extensions with custom weights for composite visualisation or analysis. These normalisation options can be previewed in real-time through the user interface before processing.

Cutout extraction implements a configurable padding factor controlling the extraction area relative to source size. All cutouts undergo resampling to a user-specified

target resolution using configurable interpolation methods including bilinear, nearest-neighbour, cubic, and Lanczos resampling to enable efficient storage as large tensors.

3. Results

We evaluated `Cutana`'s performance using *Euclid* Q1 data on 256×256 pixel cutouts from the MER catalogue. Testing was conducted on an Intel i5-13400F system with 64 GB DDR4 RAM and we compared with `Astropy`'s Cutout2D implementation. As `Astropy` focuses on single-threaded processing we use that for comparisons to establish baseline performance, utilising the parallelisation of underlying libraries via thread pinning to individual cores.

Table 1 presents the performance measurements across two test configurations varying in tile count, FITS file complexity, and source catalogue size. The primary comparison (8 tiles, 1 FITS file, 200,000 sources) demonstrates that single-threaded `Cutana` execution achieves a 2.90× improvement over `Astropy`'s memory-mapped implementation, processing 878.3 cutouts per second compared to 303.1 for `Astropy`. With four workers, `Cutana` reaches 2264.0 cutouts per second, representing a 7.47× speed-up over the baseline single-threaded `Astropy` implementation.

Table 1. Framework Comparison: `astropy` vs `Cutana` Performance

Tiles - Channels (i.e. FITS files) - Total # of sources	Tool (Threads / Workers)	Runtime (s)	Sources/sec
8 Tiles - 1 Channel - 200,000	`astropy` (1 Thread)	659.9	303.1
8 Tiles - 1 Channel - 200,000	`Cutana` (1 Worker)	227.7	878.3
8 Tiles - 1 Channel - 200,000	`astropy` (4 Threads)	626.1	319.4
8 Tiles - 1 Channel - 200,000	`Cutana` (4 Workers)	88.3	2264.0
32 Tiles - 4 Channels - 32,000	`astropy` (1 Thread)	208.0	153.9
32 Tiles - 4 Channels - 32,000	`Cutana` (1 Worker)	529.2	60.5
32 Tiles - 4 Channels - 32,000	`astropy` (4 Threads)	98.3	325.5
32 Tiles - 4 Channels - 32,000	`Cutana` (4 Workers)	69.6	460.0

The performance gains derive from: (1) vectorised batch processing eliminating per-source overhead, (2) efficient memory management through tile-based grouping, and (3) optimised I/O patterns minimising disk access latency. The batch processing approach is efficient for dense source catalogues where many cutouts originate from the same tile, as in the 8-tile configuration achieving the highest throughput rates.

`Cutana` demonstrates scaling from single to multi-threaded with a scaling factor of 2.58× comparing one to four workers for the 8 Tile Scenario, a parallel efficiency of 64.5%. The sub-linear scaling primarily results from I/O contention when multiple workers access the same FITS files simultaneously. This is noticeable as thread scaling efficiency is super-linear (7.6× sources/sec with workers) for the 32 Tiles Scenario.

The two test configurations reveal how number of sources per tile affects processing throughput. The dense configuration (8 tiles, 1 FITS file, 200,000 sources) achieves 878.3 cutouts/second with single-worker `Cutana`, demonstrating optimal performance when many sources originate from few tiles. In contrast, the sparse configuration (32 tiles, 4 FITS files, 32,000 sources) demonstrates 60.5 cutouts/second, a reduction attributable to increased tile-loading overhead. This behaviour reveals that `Cutana` operates most efficiently when compute-bound rather than I/O-bound, whilst `Astropy` exhibits limited performance variation across these scenarios.

Figure 2 illustrates the critical distinction between `Cutana`'s dynamic memory management and `Astropy`'s. With no additional memory management`Astropy` exhibits continuous memory growth, reaching 9.26 GB before completion due to per-source processing accumulating intermediate arrays. In contrast, `Cutana`'s tile-based architecture maintains bounded memory consumption regardless of catalogue size. Single-worker execution peaks at 2.49 GB, whilst four-worker parallel processing reaches 7.59

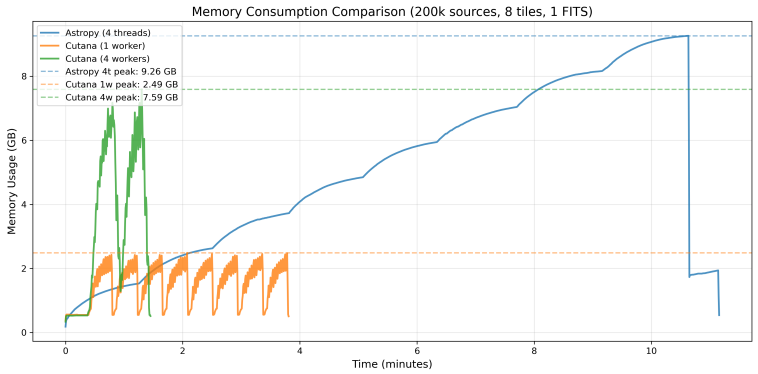


Figure 2. Memory consumption comparison processing 200,000 sources across 8 tiles. `Astropy` (blue) exhibits monotonic growth, whilst `Cutana` (orange: 1 worker, green: 4 workers) maintains bounded memory through tile-wise recycling.

GB. The characteristic sawtooth pattern in `Cutana`'s memory profile reflects batch processing cycles: each worker receives a job for ideally one tile (minimum and maximum numbers of sources can be specified to minimise overhead), processes all associated sources in batches, writes outputs, and terminates before a new one worker is spawned for the next job. This memory-efficient design proves essential for containerised deployments where resource limits are strictly enforced.

The peak memory usage of 2.49 GB per worker during tile loading phases, with average footprint of 1.8 GB during steady-state processing, enables predictable resource allocation. This prevents out-of-memory failures and maintains near-optimal throughput through intelligent load balancing.

4. Conclusion

`Cutana` addresses the critical need for scalable, memory-efficient cutout generation in the era of petabyte-scale astronomical surveys. By implementing batch processing, dynamic load balancing, and support for efficient data formats, the software tool enables efficient processing of massive datasets in state-of-the-art computational environments. The demonstrated performance improvements combined with robust memory management make `Cutana` suitable for integration into survey data processing pipelines.

Acknowledgments. We thank the *Euclid* Consortium for Q1 data access and valuable feedback. We acknowledge ESA Datalabs for computational infrastructure. Software development and manuscript preparation made use of AI tools.

References

Euclid Collaboration, Aussel, H., Tereno, I., Schirmer, M., Alguero, G., et al. 2025a, Euclid quick data release (q1) – data release overview. eprint: 2503.15302

Euclid Collaboration, Mellier, Y., Abdurro'uf, Acevedo Barroso, J. A., Achúcarro, A., et al. 2025b, A&A, 697, A1. eprint: 2405.13491

Ivezić, Ž., Kahn, S. M., Tyson, J. A., Abel, B., Acosta, E., et al. 2019, ApJ, 873, 111. eprint: 0805.2366

Miles, A., jakirkham, Stansby, D., Papadopoulos Orfanos, D., Hamman, J., et al. 2025, zarr-developers/zarr-python: v3.1.3

Navarro, V., del Rio, S., Diego, M. A., Lopez-Caniego, M., et al. 2024, in Space Data Management (Springer), 1–13

O'Ryan, D., & Gómez, P. 2025, A&A, 704, A227. eprint: 2505.03508

Astronomical Data Analysis Software and Systems XXXV
*ASP Conference Series, Vol. **VOL** Compiled: Tuesday 30th June, 2026@ 01:30*
K. Polsterer, D. Bohlender, S. Gwyn, A. Hincks and P. Teuben, eds.
*© **YEAR** Astronomical Society of the Pacific*

ZZZDEBUG!062.tex
Ehrlich, L.|textbf
Ibarra, A.
Saxton, R.
Kruk, S.
Doctor, M.
FOOBAR!conference!ADASS 2019
FOOBAR!organisations!ASP
FOOBAR! ascl:1101.010

XMMGPT: Integrating Agentic RAG and Autonomous Agents for Astronomical Data Processing

Lorenz Ehrlich[1], Aitor Ibarra[1], Richard Saxton[1], Sandor Kruk[2] and Miguel Doctor[1]

[1]*Telespazio UK for ESA, Madrid, Madrid, Spain;* lorenz.ehrlich@ext.esa.int

[2]*European Space Agency/ESAC, Villanueva de la Cañada, Madrid, Spain*

Abstract. XMMGPT is a dual purpose project which aims to serve as a unique access point to aid astronomers in their research with XMM-Newton data, and as an exploration of language models and AI systems in ESA (European Space Agency) workflows. The system is comprised of 4 main parts, a heavily customized Agentic Retrieval Augmented Generation (RAG) pipeline, a visibility checker tool, a long-term light curve generator, and an autonomous agent to execute data processing tasks from the Scientific Analysis System (SAS). Though the pipeline from user query to final answer involves an Agent no matter the specific actions required to execute the task at hand, there exists a significant difference in the architecture behind the RAG tool, and the other tools.

1. Introduction

The XMM-Newton satellite (Jansen 2001) is one of the most successful missions ever built for ESA. It has been operating as an open X-ray observatory since the beginning of 2000, producing high quality scientific results since then.

The XMM-Newton Science Operations Centre (SOC), located at ESA's European Space Astronomy Centre (ESAC, near Madrid), is responsible for the end-to-end scientific operations support of the mission, acting as the interface between the scientific community and the wider mission operations and data systems. In addition to core SOC activities such as: proposal handling, mission planning, instrument and observation monitoring, instrument calibration and provision and maintenance of the XMM-Newton Science Analysis Software (SAS; Gabriel & et al. 2004).

The SOC operates an XMM-Newton Helpdesk as the primary front door for user support. Originally the XMMGPT project was intended to serve as a Chatbot, attempting to emulate the process of the XMM-Newton Helpdesk, in which users send queries regarding SAS usage and receive back answers from experts. In order to do this, two approaches were explored; building a RAG system based on SAS documentation and Fine-tuning a language model using real world Helpdesk Tickets. This exploration resulted in the identification of major issues with building LLM-based systems for highly technical domains. In order to fine tune a language model on real world data, one typically collects question-answer pairs which serve to demonstrate the ideal response to a user query. This proved to be more difficult than assumed as real world Helpdesk tickets were much more intricate than simple questions and answers. Additionally, heavy

manual data cleaning would have been needed to remove irrelevant as well as private information that would distract the LLM during fine-tuning.

1.1. XMM-Newton RAG Pipeline

A RAG system at its core is comprised of two things: A knowledge base (KB) & a Retrieval Mechanism. A KB is a collection of documents which have been cleaned & chunked, a retrieval mechanism is any process which matches a user query to a chunk in a KB. Chunking refers to splitting parts of a document up into smaller pieces, in order to try and isolate relevant information for a given user query. A typical RAG system then takes in a user query, uses a retrieval mechanism to search chunks within the KB, and then returns a list of chunks ranked by relevance. This relevance calculation changes depending on the retrieval mechanism used.

Within XMMGPT the KB consists of the following documents: SAS User Guide, Data Files Handbook, XMM User Handbook, Pipeline Processing System, Task Descriptions, Calibration guides. Each of these documents was cleaned and irrelevant information was removed. The process of chunking documents within a KB can have large scale effects on the performance of RAG systems; Chunks too large may result in relevant information being difficult to identify, while chunks too small may result in information being scattered. While chunk size was explored with fixed intervals, the most logical and best performing strategy was semantic chunking, which involved breaking down documents hierarchically by sections, subsections etc., down to the lowest sectioned text chunk.

The retrieval mechanism involved a hybrid approach of a pure text based retriever as well as a vector embedding based retriever. BM25 was used as the pure text approach, while a fine-tuned version of *modernbert* was used for the vector embedding approach. Both retrievers gather a set of documents deemed most relevant based on internal metrics, and then ranked once more in combination using reciprocal ranked fusion. Additionally, a reranker was used to evaluate retrieved documents once more, in this case the gte-reranker-modernbert-base. In order to fine-tune the encoder used to generate vector embeddings, a synthetic training set was generated. For each individual chunk within the knowledge base, a language model was tasked with writing a question that revolves around the information found within the chunk. The language model used for this was Nous Research fine-tune of Llama3.1 8B, hosted locally. While this does not create a perfect representation of user interactions with a chatbot, it sufficed to improve retrieval performance by a noticeable margin. Additionally, 10% of the training set would serve as evaluation with which to judge different encoders, language models, chunking strategies, and rerankers.

The retrieval system also included heavy customization with respect to the nuances of the Scientific Analysis System. Within SAS there exist a variety of different tasks/commands which may be run in order to generate different scientific products. Any given scientific product thus has some tasks associated with its creation. Inside the Pipeline Processing System documentation there exists an appendix which matches each product to its associated tasks. This was used to create a sort of "naive knowledge graph" in which direct relationships from product to task were drawn. This served as an augmentation to a base retrieval approach in that, if a given product was deemed relevant to the users query, its associated tasks were searched for relevant information as an extra step. This is necessary partly because each product's description alone does not suffice in explaining the full process of its creation. Including this step provided

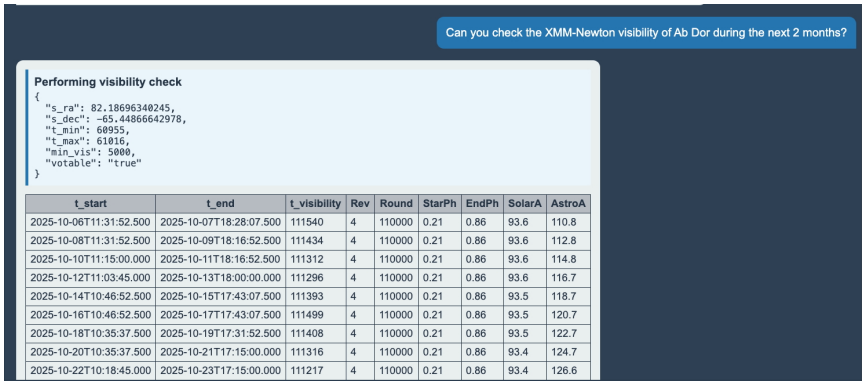
significant increases in real world retrieval performance, where questions often require information from a combination of sources within SAS documentation.

A brute-force task matching step was also implemented which consisted of gathering a set of unique tasks, and matching them to words within the user query to see whether the user had mentioned a task directly by name. In the case that a match was found, then this specific task was prioritized within the retrieval process. While this approach sounds rather brutish, added latency was negligible and performance seemed to increase, especially in regard to task-specific questions.

Query routing in the form of discretization of documents by category was also used to isolate relevant documents before searching. This means that each document type, i.e. Technical Documentation, Product based documentation etc., was assigned to a category, and a preprocessing step was implemented in which an LLM would route the user query to its appropriate category. The categories are: Installation, Product, Task, General. While in theory this step should strictly improve performance as we directly isolate information to search through to relevant documents, as aforementioned some queries necessitate the retrieval of multiple document sources, which in part inspired the inclusion of brute force task matching and "naive" knowledge graphs. Given the category Product, the system would isolate its search to the PPS documents, however its likely that the query would require task specific information. Hence, if the top retrieved document had associated tasks, then an additional search would be performed over the respective tasks.

2. XMM-Newton AI Agents

The project also explored the idea of creating LLM Agents for XMM-Newton related data processing and tasks. Agents are most simply defined as Language Models that can make decisions and use tools provided to them. While exact definitions of what an Agent is are not commonly agreed upon / do not exist, the aforementioned one shall suffice for the scope of this project. For this, there were two potential branches to examine. Branch one consisted of following a "True" Agent approach.



t_start	t_end	t_visibility	Rev	Round	StarPh	EndPh	SolarA	AstroA
2025-10-06T11:31:52.500	2025-10-07T18:28:07.500	111540	4	110000	0.21	0.86	93.6	110.8
2025-10-08T11:31:52.500	2025-10-09T18:16:52.500	111434	4	110000	0.21	0.86	93.6	112.8
2025-10-10T11:15:00.000	2025-10-11T18:16:52.500	111312	4	110000	0.21	0.86	93.6	114.8
2025-10-12T11:03:45.000	2025-10-13T18:00:00.000	111296	4	110000	0.21	0.86	93.6	116.7
2025-10-14T10:46:52.500	2025-10-15T17:43:07.500	111393	4	110000	0.21	0.86	93.5	118.7
2025-10-16T10:46:52.500	2025-10-17T17:43:07.500	111499	4	110000	0.21	0.86	93.5	120.7
2025-10-18T10:35:37.500	2025-10-19T17:31:52.500	111408	4	110000	0.21	0.86	93.5	122.7
2025-10-20T10:35:37.500	2025-10-21T17:15:00.000	111316	4	110000	0.21	0.86	93.4	124.7
2025-10-22T10:18:45.000	2025-10-23T17:15:00.000	111217	4	110000	0.21	0.86	93.4	126.6

Figure 1. XMMGPT screenshot of the visibility checker AIAgent.

In terms of XMM-Newton & SAS this would require one to build a general SAS Agent that understands and has the ability to write SAS commands & code, gather necessary observation data, resolve errors and continue doing so in a loop until the Agent

deems the task at hand complete. While this branch served as an interesting experiment to potentially explore, in part due to its nature of suiting the reinforcement learning based fine-tuning (due to its objective nature in which SAS code either returns an error or executes) of LLMs, it was decided that this would require intelligence beyond the capabilities of language models that could be hosted on local hardware. Additionally, user interactions with SAS often follow the Pareto principle in that a large bulk of usage involves a small part of possible tasks. Branch 2 offered both a simpler path to autonomous processing, and allowed for prioritization of necessary tasks, rather than the construction of a fully general agent. In this approach the system would utilize the LLM as a mediator between user and backend, exploiting the existence of several publicly available APIs, such as the XMM-Newton Upper Limit Servers (HILIGT; Saxton et al. 2021) and the Remote Interface to SAS Analysis (RISA; Gabriel & et al. 2008; Ibarra & et al. 2012), that would allow for the submission of jobs for the desired user product. This would then allow us to construct the system in such a way that the most used tasks could be prioritized and implemented at a faster pace than would be possible trying to create a general agent.

This approach has proven to be useful in several ways, while serving as a much easier to implement method than attempting to build a true agent, it has also proved to be much more reliable, cheap, and useful in unexpected ways. The use of already developed and tested tools has resulted in a much more reliable and consistent execution of tasks. Where a true agent might make errors in data collection or SAS command generation, tested APIs behave deterministically and predictably. True agents typically run in a loop in which they think, plan, resolve errors, and try different approaches, which results in the generation of a significant amount of tokens. For hosting local models this means more time spent per task, while for paid models this results in higher costs per task. Additionally, an added benefit is that one may setup the backend in such a way that multiple executions can be happening at the same time, in 2 ways. Firstly, instead of generating one light curve at a time, a user can ask for multiple sources & bands and the agent must simply extract their data individually, and then call respective APIs asynchronously.

While autonomous agent setups spend most of execution time waiting for LLM response generation, our approach instead revolves around minimal text generation and most waiting time stems from external APIs. These services respond equally fast regardless of agent setup, meaning our approach can be much faster than an autonomous agent in certain cases.

Current development is focused on a multitude of tasks. Firstly, the project is currently undergoing a major addition with the development and integration of a fully autonomous Model Context Protocol approach. While the current workflow is rigid and deterministic based on directed workflows, we believe that in due time, cheap to host language models will be capable enough to perform effectively in a fully autonomous environment. Thus for each new tool created for XMMGPT the tool is immediately also ingested into the MCP, this way we keep our main, working, approach, and may switch over to the fully autonomous setup when an LLM that meets our performance target is identified. Secondly, we are currently developing various methods to help researchers create effective proposals for XMM Newton, by way of using PIMMS as a count rate to flux converter, providing an exposure time calculator, and a pile-up estimator. Thirdly, the development of "mini-SAS" packages is under planning, which would serve two functions: Lightweight discretized packages rather than a full SAS resource when a

user is only concerned with a few specific tasks/products. And the provision of further API endpoints as SAS task executors for the XMMGPT Agent.

3. Conclusion

The project has overall resulted in the identification of several issues, bottlenecks, constraints, difficulties, benefits, and branches to explore in the further development of LLM related systems for Astronomical data processing. Namely that the fine-tuning of a language model for mission related questions and error handling can be done effectively only if either manual labor is spent cleaning up data, or data is collected in such a way that one may easily gather high-quality question-answer pairs, or informative multi-turn query-response sets. That the performance of RAG systems for this domain is heavily limited by the information density and conciseness of documentation used within the knowledge base, as well as the amount of subjects that carry over multiple documents and document types. That when constructing these naive agent type systems, latency bottlenecks between user request and data returned lie not in the language models but rather in the external API response time, which in Agent based systems is extremely beneficial. And that when working with constrained intelligence in the form of smaller, locally hosted language models, the use of external services can provide an easier method of implementation than constructing autonomous agents.

References

Gabriel, C., & et al. 2004, in ADASS XIII, edited by F. Ochsenbein, M. Allen, & D. Egret (San Francisco: ASP), vol. 314 of ASP Conf. Ser., 759

— 2008, in ADASS XVII, edited by R. W. Argyle, B. P. S., & J. R. Lewis (San Francisco: ASP), vol. 394 of ASP Conf. Ser., 183

Ibarra, A., & et al. 2012, in ADASS XXI, edited by P. Ballester, D. Egret, & N. Lorente (San Francisco: ASP), vol. 461 of ASP Conf. Ser., 643

Jansen, F. e. a. 2001, A&A, 365, L1

Saxton, R., König, O., Descalzo, M., Belanger, G., Kretschmar, P., Gabriel, C., Evans, P., Ibarra, A., Colomo, E., Sarmiento, M. H., Salgado, J., Agrafojo, A., & Kuulkers, E. 2021, Hiligt, upper limit servers i – overview. eprint: 2111.14238, URL `https://arxiv.org/abs/2111.14238`

Astronomical Data Analysis Software and Systems XXXV
*ASP Conference Series, Vol. **VOL** Compiled: Tuesday 30th June, 2026@ 01:30*
K. Polsterer, D. Bohlender, S. Gwyn, A. Hincks and P. Teuben, eds.
*© **YEAR** Astronomical Society of the Pacific*

ZZZDEBUG!064.tex
Castro, S.|textbf
Wells, A.
software
CASA! ascl:1107.013

Testing Complex Scientific Software: Key Challenges and Effective Strategies

S. Castro,[1], R. Garwood[2], A. McNichols[2], N. Schweighart[2], V. Suoranta[2], T. Tsutsumi[3], and A. Wells[2]

[1]*European Southern Observatory, Garching bei München, Germany;* scastro@eso.org

[2]*National Radio Astronomy Observatory, Charlottesville, VA, USA*

[3]*National Radio Astronomy Observatory, Socorro, NM, USA*

Abstract. Implementing a test process for a long-lived scientific software with complex dependencies and many layers of code requires a change in perspective and culture within the entire development team. It presents unique challenges that require careful strategic planning and infrastructure development. This paper describes the evolution of testing strategies for the Common Astronomy Software Applications (CASA) package, a critical tool for radio astronomy data processing. We discuss the challenges faced by the CASA team in the past decade, including heavy annual release cycles, and unclear test coverage. We then present the effective strategies that were implemented to address these challenges: redefining the development process with a dedicated verification phase, improving continuous integration tests, increasing test coverage, and collaborating with stakeholders to ultimately improve their trust in CASA. We also examine new challenges that emerged more recently, and finally we discuss how these experiences are being used to propose a test strategy for the next generation data processing system.

1. Introduction

The Common Astronomy Software Applications (CASA) package is an open-source software suite for processing single-dish and radio interferometric data (CASA Team 2022). CASA consists of a suite of applications implemented in the C++ programming language (Stroustrup 1997) and is accessible through an Interactive Python interface (Perez & Granger 2007). The origin of CASA lies in the AIPS++ project (Glendenning 1996; McMullin et al. 2006), which was started in 1992 as the successor of the Astronomical Information Processing System (AIPS) software package (Greisen 2003).

In 2004, the AIPS++ code was re-organized and migrated to CASA, and the scripting language was changed from Glish (Paxson & Saltmarsh 1993; Schiebel 1996) to Python with a layer of Python bindings known as "casapy." At the same time, the core of the AIPS++ libraries formed casacore (van Diepen & Farris 1994).

CASA is the official data processing software for ALMA, VLA, Nobeyama 45m and frequently used for other radio telescopes (EVN, GMRT, etc.).

CASA is being developed by an international consortium of scientists and software engineers based at the National Radio Astronomical Observatory (NRAO), the Euro-

pean Southern Observatory (ESO), the National Astronomical Observatory of Japan (NAOJ), and the Joint Institute for VLBI European Research Infrastructure Consortium (JIV-ERIC), under the guidance of NRAO.

The complexity of CASA—with its extensive C++ dependencies, legacy Fortran libraries, and MPI parallelization layers in both Python and C++—presents significant challenges for testing and quality assurance. Figure 1 shows a partial view of the dependency map of the CASA low-level codebase.

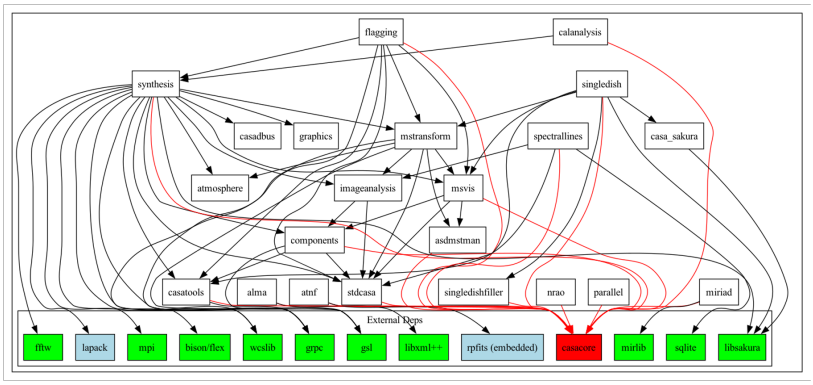


Figure 1. CASA C++ dependency map illustrating the complexity of the codebase.

Figure 2 illustrates the CASA code structure, consisting of the low-level casacore, the casatools C++ classes as the core of the interferometry and single dish algorithms, a middle layer implemented with SWIG to provide Python bindings to casatools and on top a more user-friendly interface for the end user called casatasks, which are Python scripts with interfaces defined in an XML layer. The CASA GUIs are available for visualization of the visibilities and for browsing the data.

Heuristic code is implemented in a separate repository as part of the CASA Pipelines package that uses the casatasks to access the data processing algorithms.

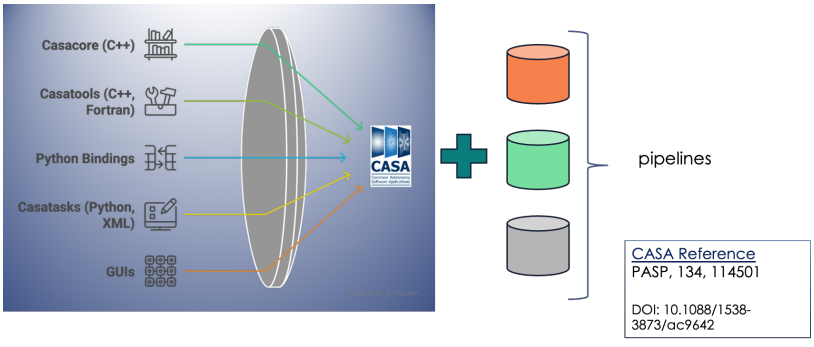


Figure 2. Overview of CASA and its pipeline packages.

Users can access CASA through an iPython shell and have access to the casatools binding functions for more advanced usage or access the casatasks and also pipeline tasks when using the CASA + Pipeline bundled packages. See the CASA (`https:`

//casadocs.readthedocs.io/en/stable/) and Pipeline documentation for more details.

2. CASA Early Challenges

The CASA development team has faced significant challenges that have impacted both development efficiency and stakeholder confidence. Below are some of the challenges identified by the team.

2.1. Heavy Annual Release Cycle

The ALMA Pipeline depends on CASA as its foundation and uses an annual CASA release for new features to support ALMA operations every year. Having a single annual release, implied that the CASA release contained a mix of stakeholders' features (not only for ALMA), infrastructure changes, maintenance bugfixes, and occasional dependency updates. This scenario scaled poorly causing many issues for the development team and stakeholders. Some identified issues were:

- Difficult to debug if issues occurred after the release, due to the large number of changes that went in the release.
- Difficult to revert to a previous release version in operations.
- High stress on the development team and stakeholders.

The accumulation of changes into a single annual release made it difficult to pinpoint the root cause of problems when they arose, fostering an atmosphere of uncertainty and frustration, especially as each release deadline approached.

2.2. Lack of Trust from Stakeholders

Validation tests by stakeholders occasionally revealed trivial bugs that should have been detected earlier in the development phase. This occurred for several reasons, including unclear information about which parts of the code had been tested during development, making it difficult to assess overall code quality. As a result, stakeholders' confidence in the software decreased, and they became more cautious about accepting changes, especially toward the end of the development cycle.

2.3. Unclear Test Coverage

While automated tests existed in a Continuous Integration (CI) system, there was no consolidated view of what was tested, when it was tested, or how it was tested. The automated test system required reorganization to provide transparency into test coverage and highlight areas needing improvement. Additionally, the main operational mode lacked test coverage, and pull request tests needed to be reviewed and extended to better cover the changes in the code before they were merged into the main branch.

3. Strategies for Testing CASA

To address the challenges described above, the CASA team implemented a set of strategies that improved both development and testing, tailoring each change to the context. The following subsections describe these approaches and the results achieved by the team.

3.1. Redefined Development Process

The first major change was to redefine the development process and introduce a verification phase at the end of each feature implementation. This phase involved a small test team and a verification lead responsible for overseeing testing during development, with the aim of improving test coverage and the quality of the code released to users.

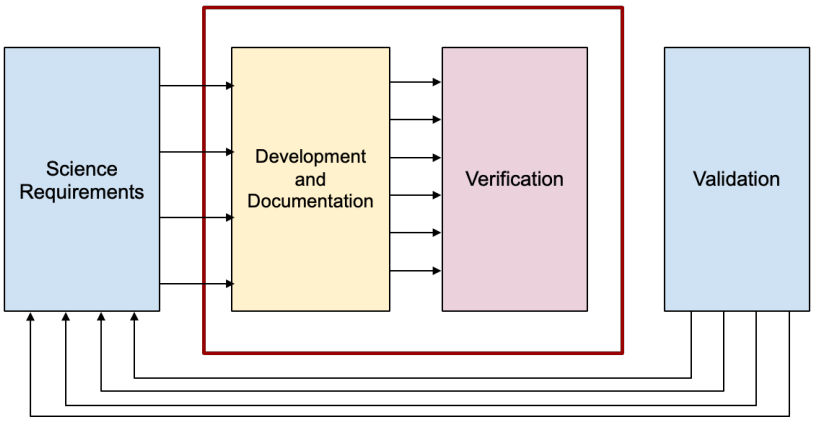


Figure 3. Redefined CASA development process highlighting a new verification phase.

3.2. Verification Phase

The verification phase was introduced as part of the development phase to increase the testing performed in feature branches and ensure that full verification would be performed before sending the code to validation. The verification team was responsible for reviewing the tests of each branch and for proposing improvements when tests were missing or not covering the changes in the code. In parallel, the test team also worked on improving the test coverage of casatasks and casatools, giving priority to those tasks with lower coverage.

Many changes were necessary to support the verification phase including infrastructure development for a tighter integration of Bamboo (Atlassian CI tool) and JIRA (Atlassian issue tracking tool), and optimization of the automated tests for branch development. The following are some of the changes applied to support the verification phase:

- Added a new Status=Verifying in the JIRA workflow, having it as a mandatory step between "Implementing" and "Validating".
- Developed a test utility package to support test development and assist the development team.

- Optimized automated tests for branch development by mapping JIRA components to the corresponding tests in the CI system.
- Added custom automation to JIRA and Bamboo that allows CASA packages to be automatically generated via checking a checkbox in JIRA.

Overall, communication needed to be improved to keep the development team informed of the changes happening in the workflow. We used the weekly meetings to share information with the developers and provided presentations to stakeholders and Users Committee on the incremental improvements happening in the system.

3.3. Improved Automated Tests for branches

Continuous integration tests were significantly enhanced and better integrated with the JIRA system. This integration provided better visibility into test results and made it easier to track which tests were associated with specific development tasks. The automated test system was optimized to execute only the tests associated to the code changes in branches. The full suite of automated tests would run as a final step of the development phase, giving a complete view of the status of the system.

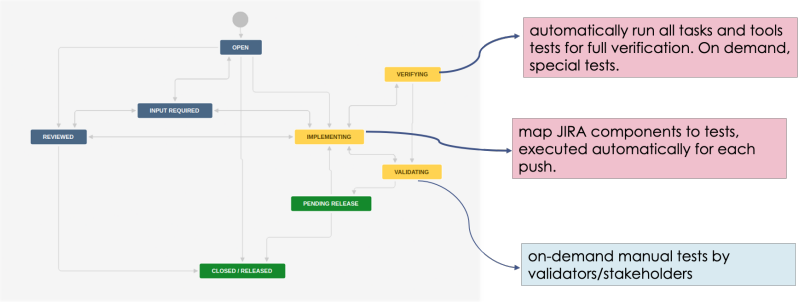


Figure 4. Figure shows the JIRA workflow with the new Verifying step.

To prevent the development team's "it works for me" problem, custom functionality was added to JIRA and Bamboo that automatically creates packages from the development branches and publishes them to JIRA once tests pass successfully. This established Bamboo as a single source of truth and ensured that consistent packages were available to stakeholders during validation testing.

3.4. Increased Test Coverage

Test coverage was systematically increased through several approaches:

- Improved visualization of test results and coverage using Allure Reports.
- Increased coverage of public APIs (casatasks and casatools).
- Added more test cases to critical tasks (prioritized test development).
- Added serial and MPI tests for parallelized imaging tasks.
- Added benchmark performance tests for main tasks using Airspeed Velocity. (ASV, `https://asv.readthedocs.io/en/latest/`)

- Added stakeholder tests to cover the most critical use-cases defined by stakeholders.

3.4.1. Allure Reports

The CASA Tests Allure Reports (`https://allurereport.org/docs/`) are the latest feature implemented by the verification team to provide a comprehensive visualization of all tests run in the CI system.

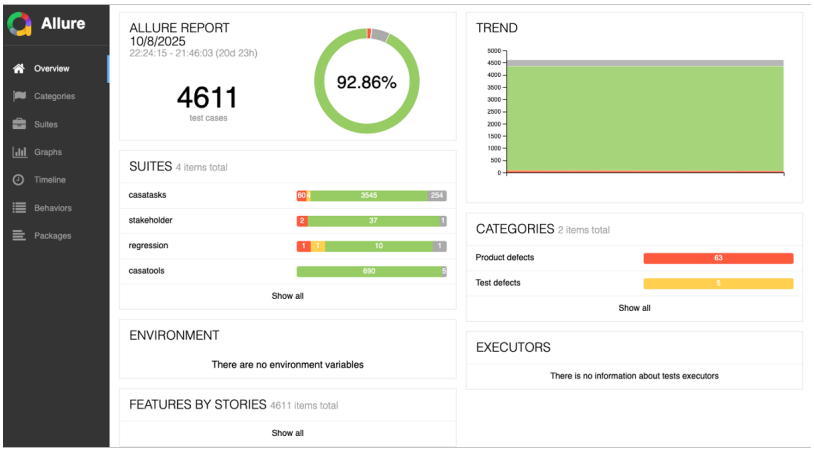


Figure 5. CASA Tests Allure Reports providing comprehensive visualization of test results and coverage.

3.4.2. CASA Benchmark Tests

The CASA benchmark tests (`https://github.com/casangi/casabench`) are available in GitHub. They use the ASV framework to provide critical insights into regression detection and performance monitoring over time. These tests help identify when code changes inadvertently impact performance, allowing the team to address issues before they reach production (See example in Figure 6). The combination of functional tests, performance benchmarks, and stakeholder-specific use cases provides comprehensive coverage of the CASA codebase.

3.5. Improved Trust from Stakeholders

The stakeholder tests represent critical use cases defined by stakeholders, who provided both data and scripts. The verification team worked closely with stakeholders to refine requirements, establish reference results for comparison, and sub-select data to reduce runtime while preserving test validity. The resulting tests were validated by stakeholders, integrated into the CI system, and used for both feature-branch and main-branch verification.

An important benefit of this approach is that side-effects of code changes for one stakeholder can be automatically verified if they affect other stakeholders, which is particularly important for common code shared across different use cases. This cross-stakeholder validation helps prevent regressions that might only be discovered much later in the development cycle. The transparency provided by these tests, combined with

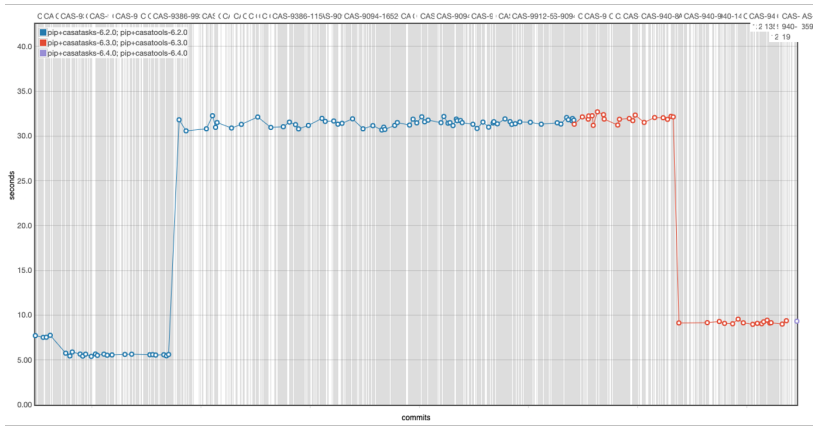


Figure 6. ASV benchmark performance tests showing an example of a regression detected in a CASA version, and the subsequent fix bringing the runtime back to normal. Y-axis is the runtime in seconds and X-axis is the CASA commits.

their integration into the CI system, has significantly improved stakeholder confidence in CASA releases.

3.6. More Frequent Releases

To avoid the accumulation of changes into a single annual release, CASA switched to quarterly releases, providing a few releases of CASA for general users during the year, while CASA+Pipeline releases remained annual for pipelines only (e.g., ALMA operations). This approach helped address the following challenges:

- Avoided mixing features and unrelated bugfixes in a single annual release.
- Provided more frequent CASA releases to non-pipeline users.
- Made pre-release packages available for download anytime.
- Reduced "release stress" in the team and stakeholders.

4. New Challenges

As the verification testing phase matured and the amount of automated tests increased, new challenges emerged that required ongoing attention and optimization. The following are some of the challenges identified:

4.1. Too Many Tests That Take Too Long to Run

The automated test suite has grown to more than 4,600 test cases, many of which are executed multiple times per build. Even with concurrent test plans and parallel jobs running in Docker when possible, the combination of limited host runners for stakeholder tests and a matrix of operating systems and Python versions has made overall test runtime a significant concern. For a single Python version, the optimized runtime for the full test suite is approximately 32 hours; a cleanup plan is therefore needed to reduce runtime and to review and remove redundant tests.

4.2. Task `tclean`: A Continuous Challenge

The `tclean` task forms images from visibilities and reconstructs a sky model. It has a complex set of parameters because it handles multiple algorithms and options. `tclean` handles continuum images and spectral line cubes, full Stokes polarization imaging, supports outlier fields, contains point-source CLEAN (Hogbom 1974) based algorithms as well as options for multi-scale and wideband image reconstruction, widefield imaging correcting for the w-term, full primary-beam imaging and joint mosaic imaging (with heterogeneous array support for ALMA). Parallelization of the major cycle is also available.

The `tclean` task presents particular challenges for testing. With 94 parameters, exhaustive parameter combination testing is computationally infeasible and not desired. Instead, the task requires "use-case testing" that captures real-world usage patterns. Stakeholder tests are particularly appropriate for this task, as they represent actual scientific workflows and data processing scenarios that combine multiple parameters in a realistic way. The complexity of `tclean` highlights the importance of stakeholder engagement in defining meaningful test cases that reflect actual scientific needs rather than theoretical parameter spaces.

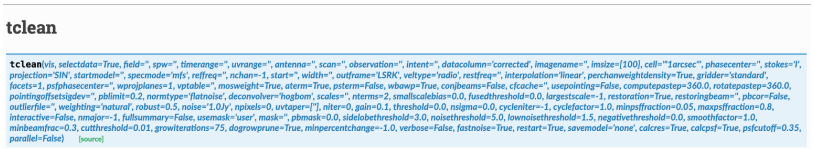


Figure 7. Task `tclean` parameter space creates difficult challenges for testing.

4.3. Build System and Coverage Reports

In its current state, coverage reporting in CASA is not fully reliable. The CASA CMake build system (`https://cmake.org/`) includes files that are not actually used, and lines from these files are still counted in the coverage reports. Addressing this issue requires refactoring the build system so that coverage metrics reflect only the lines of code that are truly in use.

5. Lessons Learned

Several important lessons have emerged from the CASA testing experience and are worth highlighting. These insights are valuable for the team and for the future maintenance of the software. Without any particular order of importance, the following are some of the lessons we learned.

- Not all tests are useful: regularly review tests, remove redundancy, and clean up the test system to keep it efficient and maintainable.
- Use coverage reports to guide test development (to add *and remove* tests).
- Provide templates for test development to improve consistency and ease of maintenance.

- Keep runtime low by optimizing the test system and reducing redundancy.
- CASA has moved to Infrastructure-as-Code for the build and test plans, which helps when adding or migrating test plans.

The experiences gained from CASA testing are now being used to define a test strategy for the new Radio Astronomy Data Processing System (RADPS), led by NRAO in collaboration with ESO and NAOJ. Starting from scratch provides an opportunity to apply lessons learned from CASA from the very beginning of the project, rather than retrofitting testing infrastructure later. Key principles being incorporated into the RADPS test strategy include:

- Start test development early, from the bottom up, by adding tests in layers.
- Track low-level mathematics in low-level code with unit tests.
- Use incremental, layered testing with unit, component, and workflow integration tests.
- Engage stakeholders early to define critical use cases for workflow testing.
- Add all tests to the CI system incrementally.
- Apply continuous integration from the start of development.
- Regularly review and clean up test suites to maintain efficiency.
- Build trust in the code from the beginning of development.

By applying these lessons proactively, the RADPS project aims to avoid many of the challenges that CASA faced in previous years, while building on the successful strategies that emerged over time.

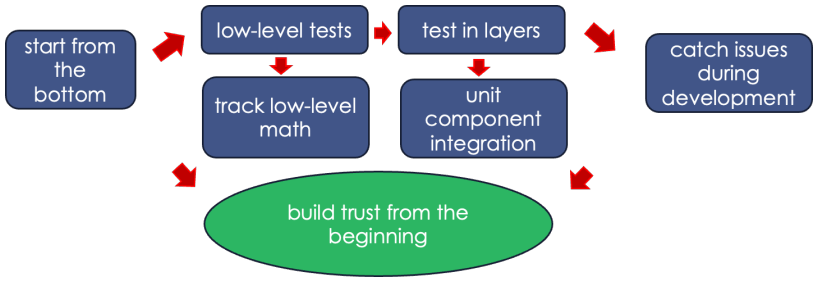


Figure 8. RADPS test strategy incorporating lessons learned from CASA testing experiences.

6. Conclusions

Testing complex scientific software requires careful strategic planning, infrastructure development, and ongoing optimization. The CASA experience demonstrates that addressing challenges through systematic improvements in development processes, verification phases, test coverage, and stakeholder engagement can significantly improve

software quality and stakeholder confidence. However, as systems grow, new challenges emerge that require continued attention, such as managing large test suites and optimizing runtime. The lessons learned from CASA are now being applied to inform the test strategy for RADPS, demonstrating the value of documenting and sharing testing experiences across projects.

Acknowledgments. The authors would like to thank the entire CASA development team and stakeholders for their contributions to improving CASA testing practices. CASA development is supported by NRAO, ESO, NAOJ, and JIV-ERIC.

References

CASA Team 2022, Publications of the Astronomical Society of the Pacific, 134, 114501

Glendenning, B. E. 1996, in Astronomical Data Analysis Software and Systems V, edited by G. H. Payne, & R. E. Lasker, vol. 101 of ASP Conf. Ser., 1

Greisen, E. W. 2003, Information Handling in Astronomy - Historical Vistas, 285, 109

Hogbom, J. 1974, Astronomy and Astrophysics Supplement Series, 15, 417

McMullin, J. P., Waters, B., Schiebel, D., Young, W., & Golap, K. 2006, in Astronomical Data Analysis Software and Systems XV, edited by C. Gabriel, C. Arviset, D. Ponz, & S. Enrique, vol. 351 of ASP Conf. Ser., 127

Paxson, V., & Saltmarsh, C. 1993, Computing Systems, 6, 1

Perez, F., & Granger, B. E. 2007, Computing in Science and Engineering, 9, 21

Schiebel, D. 1996, in Astronomical Data Analysis Software and Systems V, edited by G. H. Payne, & R. E. Lasker, vol. 101 of ASP Conf. Ser., 9

Stroustrup, B. 1997, The C++ Programming Language (Reading, MA: Addison-Wesley), 3rd ed.

van Diepen, G. N. J., & Farris, A. 1994, in Astronomical Data Analysis Software and Systems III, edited by D. R. Crabtree, R. J. Hanisch, & J. Barnes, vol. 61 of ASP Conf. Ser., 451

Astronomical Data Analysis Software and Systems XXXV
*ASP Conference Series, Vol. **VOL** Compiled: Tuesday 30th June, 2026@ 01:30*
K. Polsterer, D. Bohlender, S. Gwyn, A. Hincks and P. Teuben, eds.
*© **YEAR** Astronomical Society of the Pacific*

Automatic Data Processing with the CASA/ALMA Pipeline

Dirk Muders,[1] Kristin Berry,[2] Shawn T. Booth,[2] Chin-Shin Chang,[3,4] Hajime Ezawa,[5] Vincent C. Geers,[6] Laura Gómez,[3] John E. Hibbard,[2] Todd R. Hunter,[2] Ihor Ihnatiev,[5] Remy Indebetouw,[2] Takuma Izumi,[5] Amanda A. Kepley,[2] Devaky Kunneriath,[2] Andrew Lipnicky,[2] Ryan A. Loomis,[2] Luke T. Maud,[7] Takeshi Nakazato,[5] Marcel Neeleman,[2] Takeshi Shakunaga,[5] Patrick Sheehan,[2] Kazuhiko Shimada,[5] Kanako Sugimoto,[5] Kotomi Taniguchi,[5] Karl Torstensson,[3] Ajay Vibhute,[2] Stewart J. Williams,[6] Rui Xue,[2] Ilsang Yoon,[2] and Akira Yoshino[4]

[1]*MPIfR, Bonn, Germany;* dmuders@mpifr.de

[2]*NRAO, Charlottesville, USA*

[3]*JAO, Santiago de Chile, Chile*

[4]*University of Geneva, Geneva, Switzerland*

[5]*NAOJ, Tokyo, Japan*

[6]*UKATC, Edinburgh, UK*

[7]*ESO, Garching, Germany*

Abstract. Large astronomical facilities produce enormous amounts of data that require automatic processing to be able to analyze and publish it in a timely fashion. In this paper we will review the design and development of the automatic data processing pipeline of the Atacama Large Millimeter Array (ALMA) to achieve this goal.

From the very beginning the ALMA data products were supposed to be close to ready for scientific analysis and checked for quality. The CASA/ALMA pipeline provides processing with tailored algorithms to automatically adjust to different observing modes.

We will describe the pipeline developments, its important features in terms of heuristics and book-keeping, data product quality assessment and presentation of the results.

The pipeline is now able to process almost all standard observing mode data sets. The ongoing ALMA 2030 upgrades will bring orders of magnitude more data and the challenge to still process it in good time. But the basic concepts of the existing pipeline remain valid and will be transferred to the new system.

ZZZDEBUG!066.tex
Muders, D.|textbf
Berry, K.
Booth, S. T.
Chang, C.-S.
Ezawa, H.
Geers, V. C.
Gómez, L.
Hibbard, J. E.
Hunter, T. R.
Ihnatiev, I.
Indebetouw, R.
Izumi, T.
Kepley, A. A.
Kunneriath, D.
Lipnicky, A.
Loomis, R. A.
Maud, L. T.
Nakazato, T.
Neeleman, M.
Shakunaga, T.
Sheehan, P.
Shimada, K.
Sugimoto, K.
Taniguchi, K.
Torstensson, K.
Vibhute, A.
Williams, S. J.
Xue, R.
Yoon, I.
Yoshino, A.
observatories!ground-based
applications!CASA
applications!ALMA
Pipeline
CASA! ascl:1107.013

1. Introduction

The ALMA observatory is an international collaboration between North America (NA), Europe (EU) and East Asia (EA). It operates at mm to submm wavelengths which require extremely low atmospheric precipitable water vapor content. For this reason the high altitude (5050 m) Llano de Chajnantor plateau in the northern Atacama desert

in Chile was chosen as its location. The wavelength range would require about 10 km dish size to obtain angular resolution matched to important astrophysical processes. As this is mechanically impossible, ALMA is configured as an interferometer to synthesize the large aperture.

The synthesis process uses so called cross correlation measurements between pairs of parabolic antennas. The achievable image quality depends on how many different projected antenna-to-antenna distances (or "baselines") are sampled. Since the minimum antenna pair distance is of the order of the individual antenna diameter, smaller distances than that size would be missing. Thus ALMA uses multiple arrays with two different antenna dish diameters of 12 m and 7 m. In addition, it adds single dish / total power information from dedicated 12 m antennas.

In total there are 50 x 12 m antennas (the "Main Array"), 12 x 7 m antennas (the "Compact Array") and 4 x 12 m antennas (the "Total Power Array") (see illustration in fig. 1). The main array antennas are being relocated to different positions on the plateau using huge transporters. This further increases the selection of different baselines.

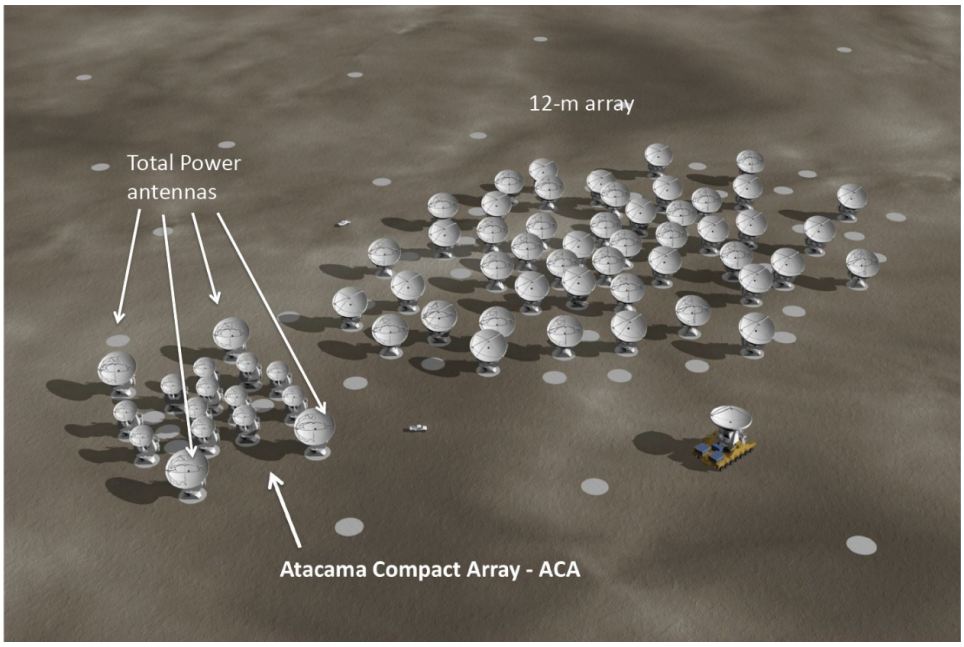


Figure 1. Sketch of the three ALMA arrays and the antenna transporter (lower right).

When the ALMA observatory was planned more than 25 years ago, one of the key requirements was an automatic pipeline to make the data products readily available for the whole community, independent of the PI's background at a particular wavelength and its corresponding processing techniques. For a radio interferometer this was fundamentally different compared to how existing facilities operated at the time.

The ALMA data rates and volumes are in the range of 1 TB per day. The upgrade to wider receiver bandwidths and better spectral resolution will lead to 10 TB per day with special setups producing 100 times more data than the current configurations. This challenge for the future processing setups is being addressed in ongoing software developments.

2. ALMA Data Processing

The processing of ALMA data uses radio interferometry and single dish techniques to calibrate, flag and image the data of these individual arrays and eventually their combination.

For interferometry, one of the major steps in calibration is to correct the signal phases for the distortion of the incoming wavefronts due to atmospheric and signal delay effects between receivers and the correlator. In addition, spectral flattening and flux calibration using secondary calibrators is applied. The calibrated data needs to be Fourier transformed to obtain images. However, the sampling of the Fourier plane is sparse, thus the inversion contains a lot of artifacts, the so called dirty beam. To overcome this problem, iterative techniques are used to detect the highest emission peaks first, replace their dirty signal with ideal Gaussians to get a clean component and compute a residual. This is repeated until the residual image RMS is at the noise level.

For single dish observations direct spectra are recorded per sky position. They are also affected by atmospheric and instrumental effects, mainly in the frequency domain, leading to structure in the *spectral* baseline (the meaning of "baseline" is unfortunately overloaded in radio astronomy). One can correct this by fitting polynomials or splines to the data while carefully excluding any frequency ranges with spectral line emission. Finally, the data is gridded to make images or cubes.

The pipeline needs to apply the necessary processing steps in an automatic way based purely on the metadata and the properties of the observed data, e.g. in terms of the phase variation or the RMS noise. All ALMA observing modes need to be detected and processed accordingly. The goal is to produce science ready data products. The ALMA pipeline has achieved this for most calibration use cases and also for standard imaging cases. More complex setups such as line forests, etc. require additional steps, which, however can still be done using pipeline commands, but with some additional parameters to steer the heuristics in a slightly different way.

3. ALMA Software Development

The ALMA software system design followed the Unified Modeling Language (UML) approach (OMG 2011). The so called "Software Science Requirements" (SSR) group, a panel of radio interferometry experts, defined the overall software requirements. A "High Level Analysis" (HLA) team helped translating those into a set of Use Cases (Lucas et al. 2007). The resulting software system structure was an observer proposal / data pipeline around archive services for meta and bulk data described by the so called "Egg Diagram" (fig. 2). It shows the life cycle of observations starting from the PI's proposal, continuing with a project, the actual scheduling of time on the telescopes, quality assessment, archiving and delivery of the data.

Several software subsystems were identified in the HLA process. The actual implementation of their functionality was delegated to many individual research institutions in the ALMA partner countries using a work breakdown structure. The pipeline subsystem was distributed from the very beginning, with developers in all of the ALMA partners' geographical regions. Online collaboration was a must, even long before video conferencing was commonly used.

In the early years from 2003 to 2009 the pipeline foundation structure was developed and tested with data from existing observatories such as the IRAM Plateau de Bure

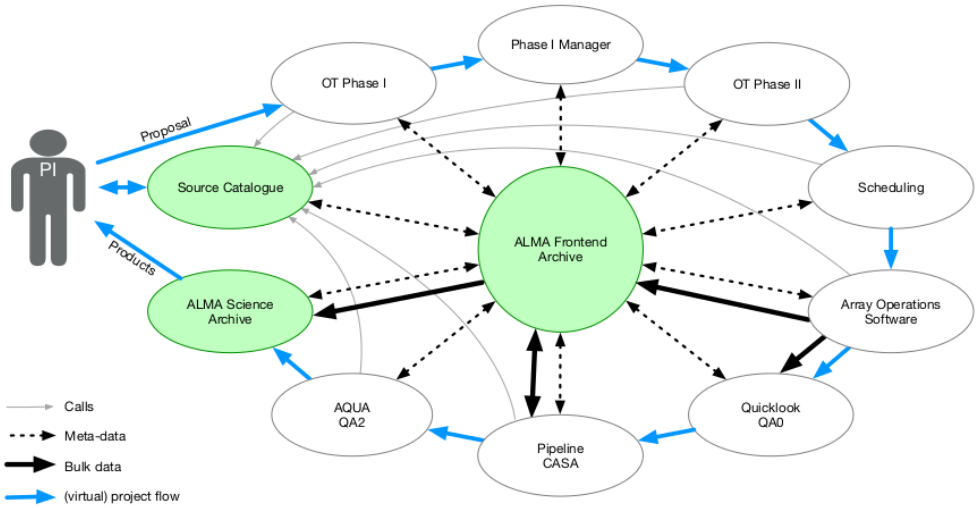


Figure 2. The ALMA software system "Egg Diagram". It shows the major high level software components which interact with several archiving subsystems. Method calls and data flow are shown.

(now NOEMA) and VLA interferometers, and the APEX 12m and IRAM 30m single dish telescopes. The underlying processing software was NRAO's AIPS++ package with the "glish" scripting language. When AIPS++ was later replaced by CASA (The CASA Team et al. 2022), the pipeline was ported to Python.

By 2009 there were working versions of the interferometry and single dish pipelines all the way to imaging. In 2011 ALMA had its "First Light" and we began tuning the pipeline heuristics to the actual data and its artifacts.

4. ALMA Data Structure

The HLA team also looked at the observing data structure, in particular with the goal to make it "pipelineable". The smallest entities would need to contain all necessary data content to be able to process it in terms of calibration, flagging and imaging, for any observing mode. This smallest set of data was called "Observing Unit Set" (OUS). The overall PI project needed to be partitioned into smaller pieces, starting with "Science Goals" which consist of one or more OUS groups, where a group has "member" OUSes, e.g. one per ALMA array (fig. 3 and 4).

The observations for a single member OUS are split into "Scheduling Blocks" (SBs) to arrive at unit scripts that take 30-60 minutes on the telescopes. The background for that time interval is that atmospheric conditions for mm/submm wavelengths can change significantly on that order of magnitude. So the scheduler has a chance to choose the right block for the current water vapor content, but also for the local sidereal time concerning the objects' elevation.

One SB may need to be observed multiple times to obtain the necessary sensitivity. Each of these runs is called an "Execution Block" (EB). This in turn leads to one ASDM (ALMA Science Data Model) dataset on disk. The ASDMs are the starting point for the

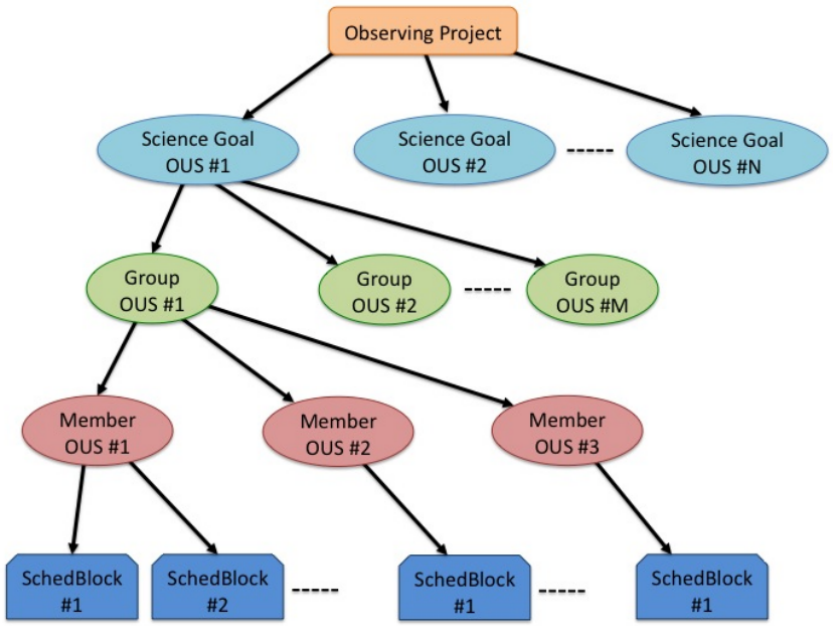


Figure 3. ALMA Science Data Model structure from Observing Project to Scheduling Block.

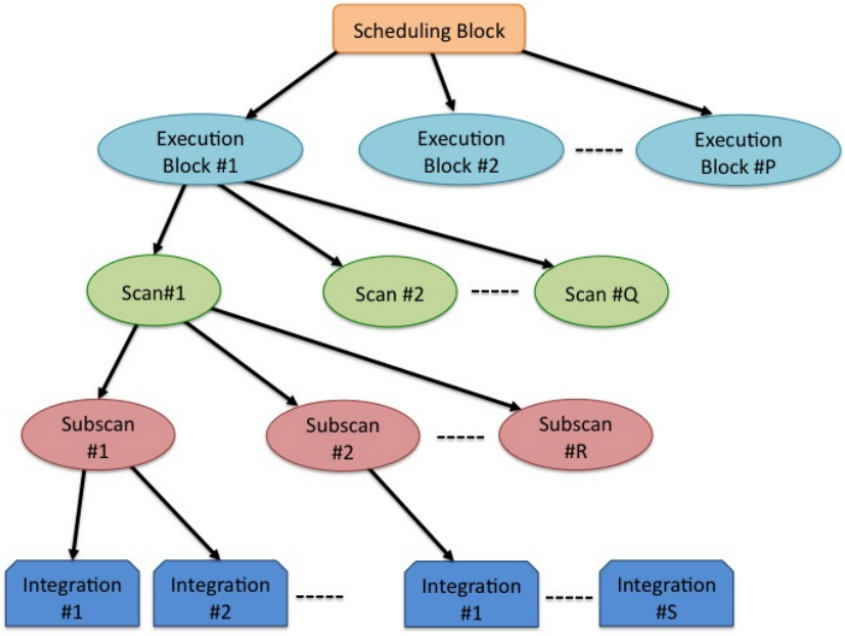


Figure 4. ALMA Science Data Model structure from Scheduling Block to Integration.

pipeline processing. In the CASA processing package each ASDM is being converted to a MeasurementSet (MS), the native CASA data format.

5. ALMA Pipeline Design

The pipeline is built on top of the CASA data reduction software. The design uses self-contained stages (also called "tasks") to perform one particular portion of the overall processing at a time. Each stage works on a standard data selection that is automatically derived from the data content. The stage heuristic to derive the processing steps is driven by data metrics, e.g. deviations from average or RMS values. Detailed descriptions of the algorithms can be found in Hunter et al. (2023).

To enable the data selection features, the ALMA pipeline maintains data proxy objects that describe the exact data content in terms of the antenna array that was used, the astronomical objects that have been observed, including coordinates and their observing intent (e.g. bandpass, amplitude or phase calibrator, science target, etc.). The proxies also store spectral window frequency and correlator polarization product information. Using them speeds up the data selection since the actual bulk data do not have to be accessed again, which would be a slow transaction.

Another key feature is the pipeline context object which carries stage results for later rendering and re-use by following stages. It is passed to every processing stage so that proxies and other information can be fetched. The algorithmic part of the stage then performs the processing and populates a results object which is finally merged into the context (see fig. 5).

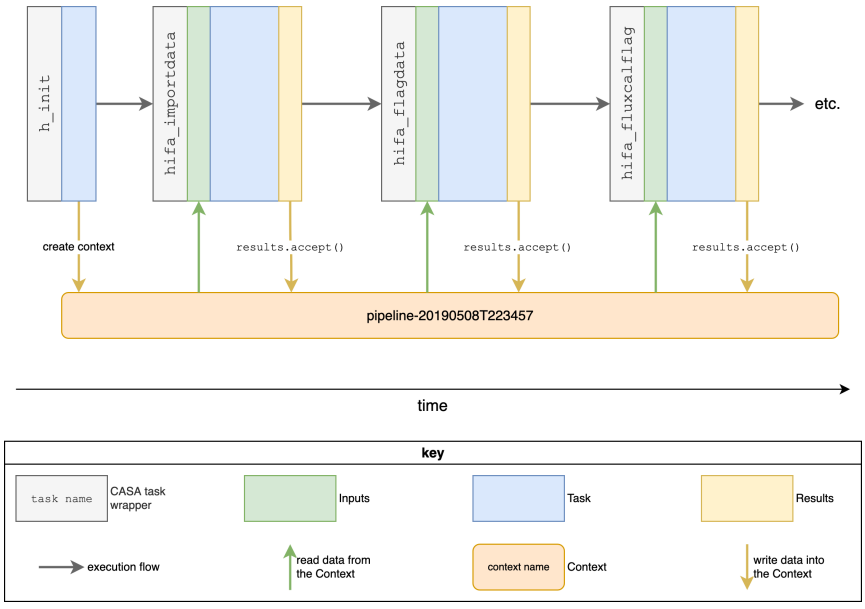


Figure 5. The pipeline context design. Data import populates domain objects that are used to derive the standard data selection per pipeline task. The task results are merged into the context for rendering and subsequent tasks.

The stage results are rendered visually in the form of HTML pages, the so called "Pipeline Weblog" (see section 8). The Weblog informs about how well the processing worked. It shows measurements such as source fluxes and achieved sensitivities, information about flagged data, various phase and amplitude corrections, and finally the

continuum images and spectral line cubes. Quality assessment scores (see section 7) are also displayed.

The pipeline products comprise calibration tables, flagging commands, images, cubes, the Weblog, and processing scripts. The PIs can retrieve them via the ALMA Archive system. PIs can re-create calibrated MSes from the ASDMs and the archived pipeline products if they want to refine the imaging steps at their own institution.

6. Processing Commands

The pipeline stages are mapped into CASA as Python commands. All pipeline command names are prefixed to distinguish them from similar CASA ones. The pipeline commands aim at being usable without any explicit parameter settings, though a number of those are still exposed to tweak the automatic algorithms. One such example are the interferometry imaging commands which take an "intent" parameter to choose whether to process certain types of calibrator sources or the scientific target. In the processing scripts multiple such commands are used, with just the intent being adjusted. All other imaging parameters such as the cell and image sizes, thresholds, and iteration parameters are computed using the actual data.

The command prefixes are hierarchically organized with `h_*` being the top level. `hif_*` stands for generic interferometry, `hsd_*` for generic single dish. Further specializations of `hifa_*` (ALMA interferometry specific), `hifv_*` (VLA(SS) specific) and `hsdn_*` (Nobeyama 45m specific) have been added over time. The latter also shows that other observatories beyond ALMA became interested in using the ALMA pipeline concept. Thus this package is currently the only existing multi-purpose radio interferometry and single dish pipeline.

7. Quality Scores

To be able to decide whether a pipeline run can be archived and delivered, it is necessary to estimate its quality. This evaluation is done via quality assessment (QA) scores. We assign QA scores per data selection that is being looped over in the stages and then aggregate these scores to compute a single stage score. To make the score comparable between stages that do very different processing steps, we settled on a normalized scheme with a number between 0.0 and 1.0, where 1.0 means "perfect" and 0.0 is "bad". The QA scores are actually entity objects that carry reason messages and data selection information to explain in more detail why a certain score has been applied.

Defining numerical ranges between 1.0, 0.9, 0.66, 0.33 and 0.0 and assigning color codes "green", "blue", "yellow" and "red" allows easy visual categorization. The additional "blue" range on top of the traditional traffic light colors was introduced to convey processing remarks that are not alarming, but point to areas that a PI might want to know. Yellow is for warnings, though the pipeline run might still be usable. Red usually means that something severe happened and that the data cannot be delivered without further manual intervention.

The stage scores are aggregated into topical categories for data handling, calibration, flagging and imaging. The aggregation algorithm is currently the minimum of all involved scores, but different approaches could be used. The overall pipeline run score is the aggregation of the topical scores.

ALMA is using the scores to implement the so called "fast lane" / "slow lane" selection process. Any pipeline run with green and blue scores is automatically passed to archiving. Runs with yellow and red scores go to manual inspection and processing.

8. Weblog

To be able to see the effects of pipeline processing stages we developed a visual feedback system called the "Weblog". It is a set of HTML pages with a hierarchical structure, starting from a top level entry page that summarizes the observational setup, the pipeline run and the datasets. The next level down is organized as "By Topic" and "By Task" pages. The topics are "datasets", "calibration", "flagging", "imaging" and "miscellaneous". This page allows jumping to the tasks that received the lowest QA scores to find problem areas.

The "By Tasks" view lists all processing steps in sequential order with short descriptions of the task goal, the QA score and a descriptive message explaining the score level. Each task offers a detail page with a longer text description, tables, plots and images illustrating the task effects and possible artifacts. A third level below these pages offers very detailed plots and dynamic filtering by data selection. Figures 6 to 8 show some example pages.

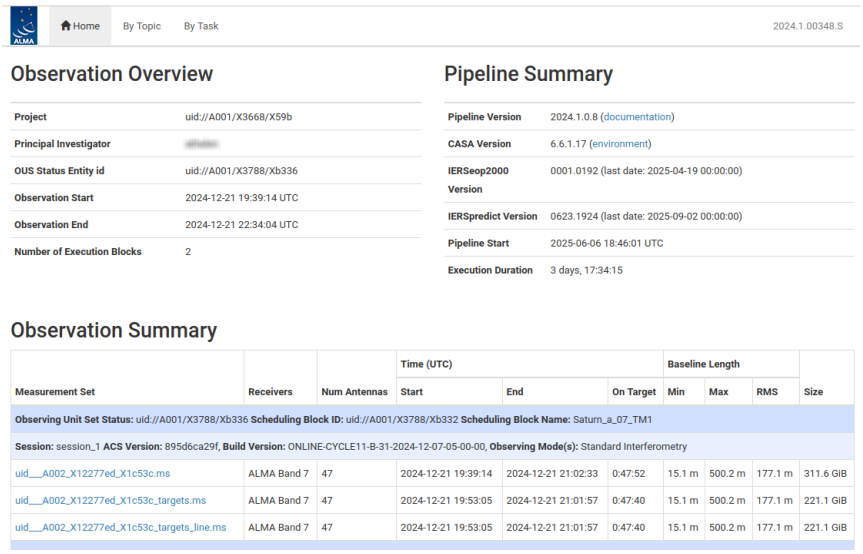


Figure 6. Top level Pipeline Weblog page.

9. Releases and Statistics

The ALMA pipeline is published at the beginning of the annual observing cycle on October 1st each year (The Pipeline Team 2025). The latest version is "PL2025", the 12th major release. More than 97% of all ALMA data is now being processed by the pipeline and the calibration portion is considered to be very stable. Imaging products can also be used directly for some projects. In other cases specific imaging is needed

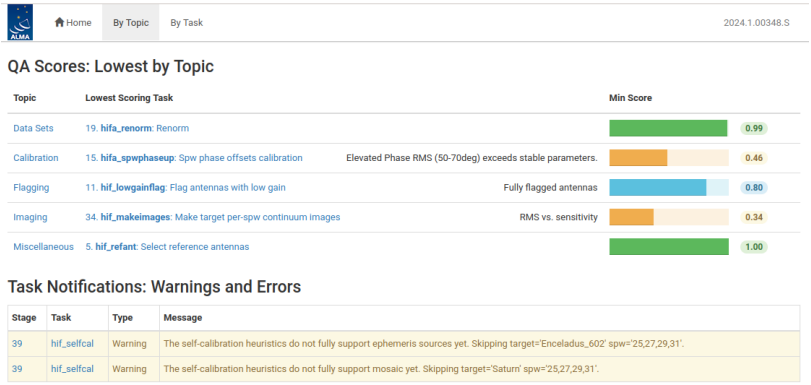


Figure 7. Pipeline "By Topic" page. The short task descriptions are links that lead to detailed results pages. QA scores are shown with color codes, numerical values and reasons.

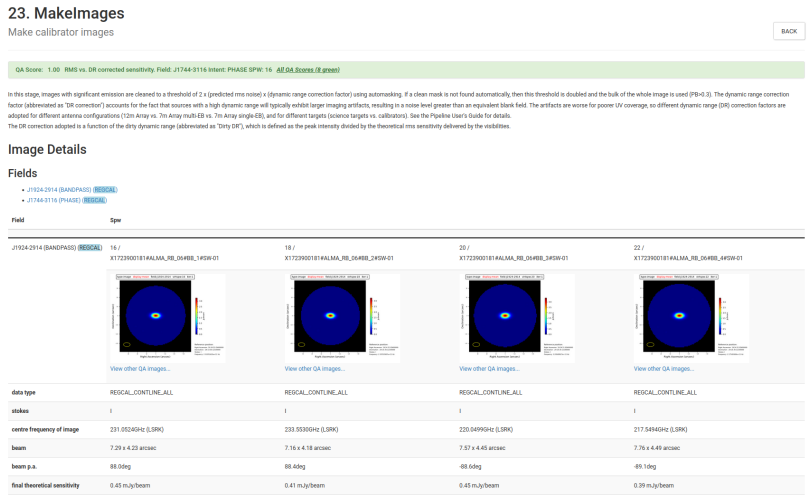


Figure 8. An interferometry imaging task detail page showing calibrator images and important metrics.

if the pipeline does not yet provide the corresponding algorithms (e.g. combination of interferometry and single dish data or multi-scale cleaning).

10. Pipeline Future

ALMA is currently working toward a big frontend and correlator improvement for the "Wideband Sensitivity Upgrade" (WSU) in the 2030s. Receiver bandwidth will double for a number of bands and the new correlator will have 80 times more channels than the current one to allow for much better spectral resolution. These changes will lead to about 100 times more data than now. Similarly, the Next Generation VLA (ngVLA) project, which also wants to keep using the pipeline for its processing, will produce 1000 times larger datasets.

The pipeline approach is still valid for the ALMA WSU and ngVLA, but the current implementation with CASA and the MeasurementSet format "version 2" limit parallelization severely to around 10-20 cores. That is why a new software packages called the "Radio Astronomy Data Processing System" (RADPS) is being developed. It aims at massively parallel processing using a new data format and a workflow approach. The goal is to speed up the processing by a factor 100 to 1000.

11. Lessons learned

During the pipeline development we learned a couple of design principles. The processing needs to be data driven. Meta data and metrics computed from the actual observations have to be used to derive algorithmic parameters that would be determined by manual data reducers. Access to the meta data via the domain proxy objects speeds up getting the inputs to all pipeline tasks without having to re-read the bulk of the MeasurementSets.

Each processing stage produces results that need to be rendered and potentially used by subsequent tasks. The pipeline run thus needs to be stateful. For the ALMA pipeline this has been solved via a context object that exists throughout the run and which is serialized at the end of it. This allows resuming and continuing a pipelne run.

To estimate the success of the processing we have developed quality assessment scores. A normalization scheme enables direct comparisons between the stages although the detailed processing is significantly different. The processing effects and products are rendered visually via a Weblog of HTML pages.

References

Hunter, T. R., Indebetouw, R., Brogan, C. L., Berry, K., Chang, C.-S., Francke, H., Geers, V. C., Gómez, L., Hibbard, J. E., Humphreys, E. M., Kent, B. R., Kepley, A. A., Kunneriath, D., Lipnicky, A., Loomis, R. A., Mason, B. S., Masters, J. S., Maud, L. T., Muders, D., Sabater, J., Sugimoto, K., Szűcs, L., Vasiliev, E., Videla, L., Villard, E., Williams, S. J., Xue, R., & Yoon, I. 2023, PASP, 135, 074501. eprint: 2306.07420

Lucas, R., Clark, B., Ezawa, H., Fomalont, E., Fuller, G., Gueth, F., Hale, A., Handa, T., Harris, G., Mangum, J., Momose, M., Muders, D., Myers, S., Nakanishi, K., Ohnishi, T., Richer, J., Salome, P., Schilke, P., Schwarz, J., Scott, S., Shepherd, D., Tatematsu, K., Testi, L., Viallefond, F., Wilson, C., & Wright, M. 2007, ALMA Software Science Requirements, ALMA Memo ALMA-70.10.00.00-002-N-SPE

OMG 2011, Unified modeling language™ (uml®). URL `http://www.omg.org/spec/UML/index.htm`

The CASA Team, Bean, B., Bhatnagar, S., Castro, S., Meyer, J. D., Emonts, B., Garcia, E., Garwood, R., Golap, K., Villalba, J. G., Harris, P., Hayashi, Y., Hoskins, J., Hsieh, M., Jagannathan, P., Kawasaki, W., Keimpema, A., Kettenis, M., Lopez, J., Marvil, J., Masters, J., McNichols, A., Mehringer, D., Miel, R., Moellenbrock, G., Montesino, F., Nakazato, T., Ott, J., Petry, D., Pokorny, M., Raba, R., Rau, U., Schiebel, D., Schweighart, N., Sekhar, S., Shimada, K., Small, D., Steeb, J.-W., Sugimoto, K., Suoranta, V., Tsutsumi, T., van Bemmel, I. M., Verkouter, M., Wells, A., Xiong, W., Szomoru, A., Griffith, M., Glendenning, B., & Kern, J. 2022, Publications of the Astronomical Society of the Pacific, 134, 114501. URL `https://doi.org/10.1088/1538-3873/ac9642`

The Pipeline Team 2025, The ALMA Science Pipeline. URL `https://almascience.eso.org/processing/science-pipeline`

Astronomical Data Analysis Software and Systems XXXV
*ASP Conference Series, Vol. **VOL** Compiled: Tuesday 30th June, 2026@ 01:30*
K. Polsterer, D. Bohlender, S. Gwyn, A. Hincks and P. Teuben, eds.
*© **YEAR** Astronomical Society of the Pacific*

ZZZDEBUG!067.tex
Weilbacher, P. M.|textbf
Vidante, C.
Wendt, M.
Martens, S.
Hayes, M.
Adamo, A.
Castro, N.
Dreizler, S.
Jeanneau, A.
Kelz, A.
Richard, J.
data!simulated
data!reduction
data!analysis
data!cube

BlueMUSE: science software for Europe's next large integral field spectrograph

Peter M. Weilbacher[1], Cinta Vidante[1], Martin Wendt[2], Sven Martens[3], Matthew Hayes[4], Angela Adamo[4], Norberto Castro[1], Stefan Dreizler[3], Alexandre Jeanneau[5], Andreas Kelz[1], and Johan Richard[5]

[1]*Leibniz-Institut für Astrophysik Potsdam (AIP), Potsdam, Germany;* pweilbacher@aip.de

[2]*Institut für Physik und Astronomie, Universität Potsdam, Germany*

[3]*Institut für Astrophysik und Geophysik, Universität Göttingen, Germany*

[4]*Department of Astronomy, Stockholm University, Sweden*

[5]*Centre de Recherche Astrophysique de Lyon (CRAL), Saint Genis Laval, France*

Abstract. BlueMUSE is a new blue-optimized, medium spectral resolution, panoramic integral field spectrograph being developed for ESO's VLT. While building on the legacy of the much requested MUSE instrument, its blue wavelength coverage to the atmospheric cutoff (~ 350 nm) together with a higher spectral resolution ($\overline{R} > 3500$) and a $1' \times 1'$ field will make it unique. In Galactic, extragalactic, and high-redshift domains, BlueMUSE will enable new science not possible with other instruments.

Here we present our work regarding science software that is developed together with the instrument to carry out simulation, reduction, and analysis of data. We describe BlueSi, the simuation tool to create realistic mock cubes of BlueMUSE. We also simulate the expected raw data of the instrument, and describe the state regarding calibration data and its use in pipeline testing and plans for simulating on-sky exposures. We present plans for the data reduction pipeline, derived from the successful MUSE processing software. Finally, we briefly present ideas for a comprehensive data analysis software environment for automated analysis of the datacubes produced by BlueMUSE.

1. The BlueMUSE instrument

The MUSE integral-field spectrograph (Bacon et al. 2010) is one of the most successful astronomical ground-based instruments so far (Roth 2024). It has been operating at the ESO Very Large Telescope (VLT) since 2014, with consistently high oversubscription factors, leading ESO-based publications since 2020. It is, however, lacking capabilities to observe the blue end of the optical wavelength range, and has too low spectral resolution for a number of science cases. Therefore, the BlueMUSE instrument was developed, to supplement the suite of instruments available at the VLT and fill this gap.

Updating the original BlueMUSE White Paper (Richard et al. 2019), Richard et al. (2024) recently gave an update on the plans for the instrument and the science that determines its design, which encompasses topics from the Solar System and the Milky

Way to objects in the distance universe. Key science cases were identified that are concerned with massive star evolution, the interstellar medium and HII regions in extreme starburst galaxies, and gas flows around galaxies in deep fields in the redshift range $2 < z < 3$. Taking these topics into account, the instrument will feature a $1' \times 1'$ field of view, sampled with rectangular spaxels of $0\farcs2$ to $0\farcs3$ size, providing a wavelength range 350 to 580 nm with a spectral resolution of $\overline{R} > 3500$, and a throughput larger than 26% and at least 15% at the blue end.

2. The Science Software

The science software consists of the raw data and cube simulation tools, a data reduction pipeline, and a data analysis package. Other software like an exposure-time calculator exist but are ignored in the following.

2.1. Cube simulation

The BlueSi software to simulate datacubes was already documented by Wendt et al. (2024). It is an end-to-end simulation software that is updated with instrument parameters and takes a scene configuration file to define the objects in the field of view. Currently, scenes can be star clusters and high-redshift Lyman-α emitters. Work to implement the gas content of young star clusters and H II regions as well as a galaxy filling the field of view is ongoing. Together with a simulation configuration that defines atmospheric conditions and exposure time, BlueSi generates fully synthetic datacubes, including instrumental and observational effects. The modular and configuration-driven design allows systematic variation of instrument parameters and observing conditions, enabling reproducible performance studies.

Such datacubes are used to study the impact of instrument parameters and observing setup on scientific measurements, such as the accuracy of derived stellar parameters or detection limits for high-redshift galaxies.

While BlueSi operates at the datacube level for science simulations, complementary simulations are used to model detector-level raw data.

2.2. Raw data simulation

BlueMUSE will split the field of view into 16 horizontal subsections and then further use integral field units (IFUs) with their image slicers on each sub-field to split the light up into 48 elements. Light from these slices then passes through a spectrograph to be recorded on 4k×4k CCDs, one for each of the 16 IFUs. Simulating raw data has to take this into account. However, we assume for now that difference between spectrographs will be small and so only simulate data on one CCD. This is sufficient to simulate calibration exposures with uniform illumination, but will need to be revisited when creating mock science exposures with spatially varying content. The up-to-date optical model of the BlueMUSE spectrographs is used as basis for modeling distortions on the detector and assigning wavelengths to pixels. Typical read-out noise is taken from existing CCDs and modeled as a constant, poissonian photon noise is added to the exposures after computing the illumination on the detector. For the PSF we use the width specified for the instrument.

With this process we successfully created bias frames and lamp flat exposures as well as various arcs, including a Fabry-Perot etalon. We were able to reduce the arcs

and create wavelength calibration frames. Because wavelength calibration in the blue optical range is a concern, our principal aim so far was to study how the selection of different arc lamps and combination of several of them influences the residuals of the wavelength calibration against the truth that we know from the optical model.

2.3. Data reduction

The data will be reduced by a largely automated system within the ESO framework, i.e., using C/CPL for the reduction modules (Banse et al. 2004), linked together by scripting (e.g., EDPS Freudling et al. 2024). This pipeline will transform detector-level raw data into datacubes. Like the rest of the instrument, its design will heavily lean on the MUSE pipeline, described in detail by Weilbacher et al. (2020).

One of the changes will be related to the handling of rectangular spaxels which will exist in BlueMUSE data. The idealized unrotated square that was still a good-enough approximation for MUSE data will no longer be accurate enough for the derivation of drizzle-like overlaps. Instead, the full spatial polygon needs to be taken into account (as also discussed by e.g. Law et al. 2023). For BlueMUSE the increased processing time for this more complex computation will be approximately compensated by the reduced data size.

Other improvements and changes studied are the propagation of the line profile (LSF), better flat-fielding (both discussed in Weilbacher et al. 2022), and an easier and more accurate geometrical calibration. Combinations on the cube-level and of MUSE and BlueMUSE data are another topic of the current design study. The calculation of covariances for the final resampling step is another topic currently under investigation. All scientific cases would benefit from more accurate uncertainty estimates. While we are able to derive the full covariance information for any datacube, their application during data analysis is cumbersome and slow due to their size. We currently investigate approaches that approximate the covariances to address this issue. In addition, we are also looking into the possibility to create tools that enable the efficient handling of the full covariance information in post-data reduction steps.

2.4. Data analysis

The data analysis software (DAS) is planned to start from reduced datacubes. It will provide automated analysis for a variety of scenes and thereby enable statistical studies and the exploration of vast datasets. Examples of the scenes would be resolved galaxies and extragalactic deep fields where different objects within a field would be classified. A goal will be production and archiving of high-level data products from all data as soon as they become public.

The structure of the DAS will consist of four main modules: (1) source identification, (2) extraction, (3) classification and redshift or velocity determination, and (4) cataloging. An optional fifth module might provide options for visualization and manual identification. The modules are connected through a database which contains configuration and classification parameters as well as user catalogs. Each module reads the input from this database and stores its output in the database as well. The final output are the advanced data products, i.e., objects which can directly be used in publications.

3. Outlook

The BlueMUSE instrument will become the next panoramic IFS at the VLT. It is based on the MUSE instrument and will have a similar spatial coverage but will feature optimized coverage of the blue wavelength range down to the atmospheric cutoff and a higher spectral resolution. Various other improvements will be implemented in hardware. BlueMUSE just started the design phase and can be expected to be at the VLT in the early 2030s.

The BlueMUSE Science Software encompasses four parts:

- Raw data simulation software can so far prepare calibration exposures and will be enhanced to also provide complex on-sky data.
- The data reduction software will be based on the existing MUSE software package but will be adapted and improved in a number of ways, most notably to handle rectangular spaxels, and propagate covariance information.
- The BlueSi cube simulation code can already produce datacubes with various atmospheric and instrumental effects as if the data were reduced by the reduction pipeline.
- The data analysis will be done with a new comprehensive software package to automate analysis tasks for a variety of science to be carried out with BlueMUSE.

All of them will be further developed in the next years and extensively tested to be ready as soon as BlueMUSE starts taking data.

Acknowledgments. PMW, CV, and SM were partially supported by the BMFTR through the ErUM program (VLT-BlueMUSE 05A23BAC and 05A23MGA). NC acknowledges funding from the Deutsche Forschungsgemeinschaft (DFG), CA 2551/1-2.

References

Bacon, R., et al. 2010, in Ground-based and Airborne Instrumentation for Astronomy III, vol. 7735 of SPIE Conf. Ser., 773508

Banse, K., et al. 2004, in Astronomical Data Analysis Software and Systems XIII, edited by F. Ochsenbein, M. G. Allen, & D. Egret, vol. 314 of ASP Conf. Ser., 392

Freudling, W., Zampieri, S., Coccato, L., Podgorski, S., Romaniello, M., Modigliani, A., & Pritchard, J. 2024, A&A, 681, A93

Law, D. R., E. Morrison, J., Argyriou, I., Patapis, P., Álvarez-Márquez, J., Labiano, A., & Vandenbussche, B. 2023, AJ, 166, 45

Richard, J., et al. 2019, arXiv, arXiv:1906.01657

— 2024, in Ground-based and Airborne Instrumentation for Astronomy X, edited by J. J. Bryant, K. Motohara, & J. R. D. Vernet, vol. 13096 of SPIE Conf. Ser., 1309622

Roth, M. M. 2024, Research Notes of the American Astronomical Society, 8, 54

Weilbacher, P. M., Palsa, R., Streicher, O., et al. 2020, A&A, 641, A28

Weilbacher, P. M., et al. 2022, in Software and Cyberinfrastructure for Astronomy VII, vol. 12189 of SPIE Conf. Ser., 1218912

Wendt, M., Castro, N., Martens, S., Pharo, J., Weilbacher, P. M., Krajnović, D., & Richard, J. 2024, in Software and Cyberinfrastructure for Astronomy VIII, edited by J. Ibsen, & G. Chiozzi, vol. 13101 of SPIE Conf. Ser., 1310123

Astronomical Data Analysis Software and Systems XXXV
*ASP Conference Series, Vol. **VOL** Compiled: Tuesday 30th June, 2026@ 01:30*
K. Polsterer, D. Bohlender, S. Gwyn, A. Hincks and P. Teuben, eds.
*© **YEAR** Astronomical Society of the Pacific*

ZZZDEBUG!071.tex
Castaldini, L.|textbf
Bulgarelli, A.
Falco, R.
Parmiggiani, N.
De Rosa, A.
Aboudan, A.
Amati, L.
Calabretto, P.
Calzolari, F.
Di Piano, A.
Fioretti, V.
Marinoni, A.
Mustafa, G.
Panebianco, G.
Rizzo, A.
Spadoni, L.
Virgilli, E.
Earth!atmosphere
Artificial intelligence
algorithm!machine learning!deep learning
observatories!space-based
astronomy!gamma-ray
satellites
libraries!HDF5
data!pipelines
pipelines
packages!Docker
computer languages!C++
computer languages!C++11
computer languages!Python
data!pipelines!processing
data!analysis!real time (RTA)
data formats!HDF5
tensorflow
software!framework!rtapipe
rate! ascl:1904.014
pile-up! ascl:1806.014

GammaSky: Edge AI Computing for X- and Gamma-Ray Phenomena in real-time

L. Castaldini[1], A. Bulgarelli[1], R. Falco[1], N. Parmiggiani[1], A. De Rosa[1], A. Aboudan[2], L. Amati[1], P. Calabretto[1], F. Calzolari[3], A. Di Piano[1], V. Fioretti[1], A. Marinoni[3], G. Mustafa[1,4], G. Panebianco[1], A. Rizzo[5], L. Spadoni[4], and E. Virgilli[1]

[1]*INAF/OAS Bologna, Via P. Gobetti 93/3, I-40129 Bologna, Italy.*

[2]*CISAS "G. Colombo" - Università di Padova, Via Venezia 1, I-35131 Padova, Italy.*

[3]*ISAC Bologna, Via P. Gobetti 101, I-40129 Bologna, Italy.*

[4]*Alma Mater Studiorum - Università di Bologna - Via Zamboni 33, I-40126 Bologna, Italy.*

[5]*INAF/OA Catania, Via Santa Sofia 78, I-95123 Catania, Italy.*

Abstract. The GammaSky project investigates the use of artificial intelligence (AI) on edge computers for real-time detection and analysis of high-energy phenomena such as Gamma-Ray Bursts and Terrestrial Gamma-ray Flashes (TGFs). The experiment operates at the "O. Vittori" Observatory on Monte Cimone (Modena, Italy), observing TGFs in real conditions. The setup combines a fast NaI(Tl) scintillator detector coupled to a SiPM readout with a RedPitaya STEMlab 125-14 based acquisition board and an NVIDIA Jetson Orin Nano for low-power real-time processing. A dedicated C++ Real-Time Analysis pipeline manages waveform streaming, machine-learning event reconstruction and detection. GammaSky is a testbed for deploying AI in constrained edge environments representative of space applications, enabling efficient onboard data reduction and autonomous science alert generation for future gamma-ray observations.

1. Introduction

A growing trend in space applications is onboard AI, which addresses the challenge of moving AI inference onboard (Izzo et al. 2022). The GammaSky Project investigates real-time analysis of high-energy gamma-ray transients, such as Gamma-Ray Bursts and Terrestrial Gamma-ray Flashes (TGFs) and Gamma-ray Glows (Wada et al. 2019) using AI on edge devices. Building on the heritage of the AGILE mission (Tavani et al. 2008) and the Gamma-Flash project (Calabretto et al. 2025), it aims to develop onboard AI platforms for future space applications using high energy atmospheric events as a testbed. Deployed at the "O. Vittori" Observatory (CNR-ISAC, Monte Cimone, 2165 m elevation),[1] GammaSky is designed to advance signal reconstruction, high-energy tran-

[1]https://cimone.isac.cnr.it/

sient detection, and spectral characterization of real atmospheric gamma-ray events. This contribution presents the current experimental setup that brings real-time onboard AI data processing closer to operation.

2. Experiment Setup

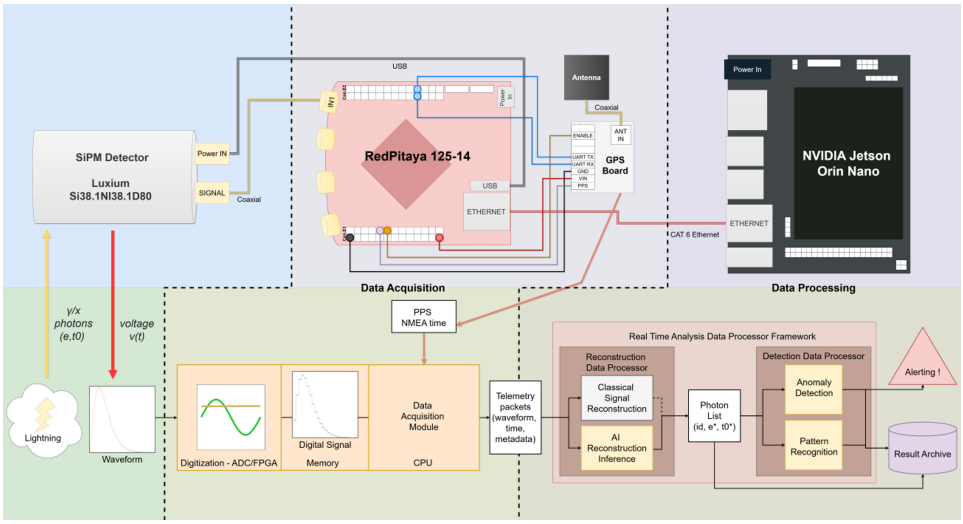


Figure 1. A simplified view of the GammaSky Hardware and Software Setup.

The GammaSky hardware is composed of three main components (Figure 1): the detector, the acquisition board, and the AI processing board. The detector is a Luxium Solutions SiPM scintillation unit with a $1.5'' \times 1.5''$ NaI(Tl) crystal. We adopted SiPM-based front-end electronics over PMTs because it reduces power and volume, offers higher efficiency and lower noise, supports low-voltage operation, and is insensitive to magnetic fields. Such properties improve SNR and waveform stability, advantageous for training and real-time ML inference on edge hardware. The fast front-end produces pulses of a few microseconds, suitable for time-resolved studies of atmospheric transients. To digitize these signals, we use the Red Pitaya STEMlab 125-14, based on a Xilinx Zynq-7000 SoC with dual-core ARM Cortex-A9, FPGA fabric, and a 14-bit 125,MSps ADC; a GPS add-on provides PPS (Pulse Per Second)-disciplined absolute timing with microsecond resolution, crucial for correlating TGFs with lightning. For onboard inference, we adopt the NVIDIA Jetson Orin Nano, a commercial off-the-shelf platform of growing interest for in-orbit AI; its Ampere GPU (1024 CUDA, 32 Tensor Cores) lets us run diverse ML/DL models on the same edge device and compare latency/throughput within a $\sim$15 W power budget. The SiPM is powered via the Red Pitaya's USB port, and its analog output is connected to one RF input for digitization by the onboard ADC. The Red Pitaya handles waveform digitization and GPS-based time-tagging, while the Jetson performs real-time AI processing of the acquired data. The Gammasky Experiment has been installed at the "O. Vittori" Observatory in June 2025. The acquisition board and the detector are installed inside a dome on the terrace, as shown in the Figure 2, having access to a vast portion of the sky. The process-

ing board, instead, is deployed inside the computing centre, where the environment's temperature and humidity are controlled to increase component life.

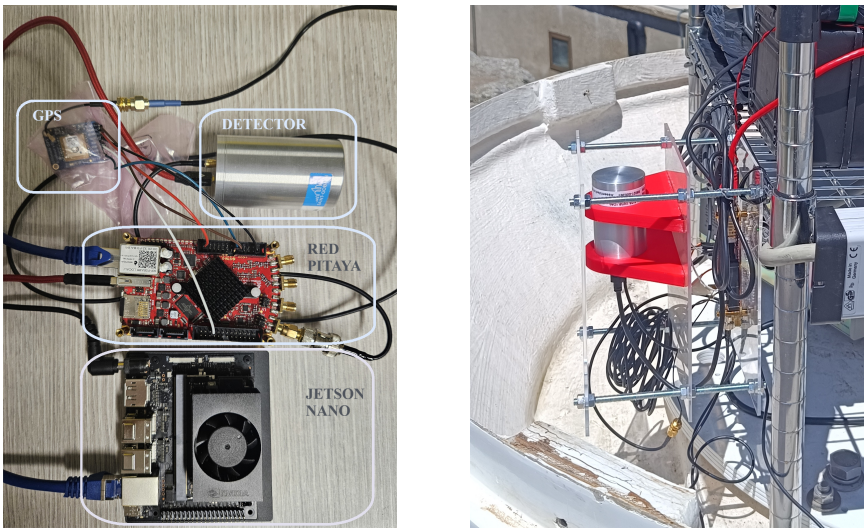


Figure 2. *Left:* The GammaSky's hardware components. *Right:* The detector and data acquisition board installed, inside a dome of the "O. Vittori" Observatory

3. Real-Time Analysis Pipeline

The ML operations in GammaSky serve two goals: (i) *event energy reconstruction* from digitized waveforms, and (ii) *anomaly detection* on reconstructed photon counts to identify short transients; such objectives drive the software architecture design. On the Red Pitaya, the FPGA is configured as an oscilloscope with a selectable trigger; it records time windows of 16 384 samples at 8 ns resolution from the detector. A Data Acquisition Module (DAM) (Bulgarelli et al. 2023) configures the FPGA, reads raw buffers, and assigns absolute timestamps using GPS messages and the 1 PPS signal to bound oscillator drift. The DAM streams the digitized waveforms to the Jetson over TCP/IP using ZeroMQ.[2] On the Jetson, the Real-Time Analysis Data Processor (RTADP) framework (Bulgarelli et al. 2025) has been ported to C++ and *supports pipeline integration*, enabling ingest, buffering, preprocessing, and AI inference. Results are written to HDF5[3] files. For onboard inference, we use LiteRT[4] together with XNNPACK (optimized CPU kernels), enabling efficient execution of floating-point and quantized models on resource-constrained hardware. The analysis pipeline consists of data processors operating sequentially. The first stage performs waveform-based reconstruction to estimate event energy and time, producing structured datasets (photon lists). The second stage performs time-series analysis of the reconstructed counts to

[2]Lightweight messaging library for high-throughput, low-latency transport: `https://zeromq.org/`

[3]Hierachical Data Format: `https://www.hdfgroup.org/solutions/hdf5/`

[4]Formerly TensorFlow Lite runtime: `https://ai.google.dev/edge/litert`

identify patterns of increased activity. Custom workers allow each stage to execute user-defined algorithms or DL models. InfluxDB provides real-time monitoring of event rates and alarms for anomalous acquisition conditions or hardware errors. The preliminary pipeline setup described in (Spadoni 2025) supports continuous real-time processing with substantial performance margins. Integration tests on the Jetson Orin Nano demonstrate sustained processing rates above 2 kHz using a lightweight test ML model for waveform area estimation, with an average inference latency of ∼0.45 ms per waveform, well above the nominal acquisition rate of ∼200 events s^{-1}. The whole pipeline runs on a single CPU core without GPU acceleration, keeping memory usage below 1 GB and power consumption around 5 W.

4. Conclusions

We presented GammaSky as a hardware and software architecture for a technological demonstrator of real-time edge AI gamma-ray detection and analysis, combining a SiPM-based NaI(Tl) detector, a GPS-disciplined Red Pitaya for waveform acquisition, and an NVIDIA Jetson Orin Nano for onboard AI. The proposed software chain supports two machine-learning tasks aligned with the scientific use case: energy reconstruction from digitized waveforms and anomaly detection on reconstructed photon-count time series, implemented within a modular RTADP-based pipeline providing deterministic data flow, microsecond-level timing, and low-power operation compatible with CubeSat-class constraints. GammaSky is conceived as a flexible experimental platform to host and evaluate reconstruction and detection algorithms under realistic operating conditions, including field operation at 2165 m elevation and lightning correlations. In this context, the primary contribution of this work is the definition and experimental assessment of the end-to-end acquisition and real-time processing architecture. Preliminary integration tests show that the pipeline sustains processing rates above 2 kHz on edge hardware, with ∼0.45 ms inference latency, sub-GB memory usage, and a few watts of power consumption, indicating suitability for real-time inference with light ML models and margin for increased complexity at the instrument event rate. Future work will focus on developing data-analysis models operating on reconstructed events as a next step toward scientific exploitation, along with further optimization of the processing pipeline to improve inference performance. In parallel, reconstruction models will be refined to deal with pile-up scenarios and benchmarked against classical algorithms under field conditions, enabling a quantitative assessment of performance.

Acknowledgments. We gratefully acknowledge INAF to support the development of this project with a Mini Grant 2024. We acknowledge ISAC-CNR institute to host our experiment at the "O. Vittori" observatory.

References

Bulgarelli, A., et al. 2023, arXiv e-prints, arXiv:2302.08321
— 2025, arXiv e-prints, arXiv:2511.03760
Calabretto, P., et al. 2025, in Journal of Physics Conference Series (IOP), vol. 2985, 012015
Izzo, D., et al. 2022, arXiv e-prints, arXiv:2212.06662
Spadoni, L. 2025, Master's thesis, Università di Bologna. eprint: amslaurea.unibo.it:37596
Tavani, M., et al. 2008, Nuclear Instruments and Methods in Physics Research A, 588, 52
Wada, Y., et al. 2019, Communications Physics, 2, 67

Astronomical Data Analysis Software and Systems XXXV
*ASP Conference Series, Vol. **VOL** Compiled: Tuesday 30th June, 2026@ 01:30*
K. Polsterer, D. Bohlender, S. Gwyn, A. Hincks and P. Teuben, eds.
*© **YEAR** Astronomical Society of the Pacific*

ZZZDEBUG!072.tex
Kent, B. R.|textbf
Kagan, S.
Sheckard, J.
Masters, J.
Kern, J.
Haynes, M. P.
archives!multiple!National Radio Astronomy Observatory (NRAO) data archive

Enhancing Access to Legacy Radio Datasets: The ALFALFA Survey in the NRAO Archive

Brian R. Kent,[1] Sam Kagan,[1] James Sheckard,[2] Joseph Masters,[1] Jeff Kern,[1] and Martha P. Haynes[3]

[1]*National Radio Astronomy Observatory, Charlottesville, VA, USA;* bkent@nrao.edu

[2]*National Radio Astronomy Observatory, Socorro, NM, USA;*

[3]*Cornell Center for Astrophysics and Planetary Science, Space Sciences Building, Cornell University, Ithaca, NY, USA*

Abstract. The NSF National Radio Astronomy Observatory (NSF NRAO) is expanding its scientific data archive to include legacy radio astronomy survey products, beginning with the ingestion of neutral hydrogen (HI) spectral-line data cubes from the Arecibo Legacy Fast ALFA (ALFALFA) survey. Conducted with the 305-m Arecibo radio telescope from 2005-2012 with its seven-beam ALFA receiver, ALFALFA represents one of the most extensive extragalactic HI surveys ever undertaken, covering more than 7,000 square degrees of sky and cataloging over 30,000 galaxies. By curating the ALFALFA data cubes within the NRAO archive, the project ensures long-term preservation through standardized metadata and scalable storage solutions. Integration into the NRAO archive ecosystem enables uniform data discovery and interoperability with widely used astronomical analysis platforms. Scientifically, this effort enhances opportunities for new research into galaxy evolution, the HI mass function, and local Universe cosmology, while enabling cross-facility investigations that combine ALFALFA products with observations from NRAO facilities. This initiative both safeguards the scientific legacy of important large surveys and amplifies the long-term impact of HI surveys for the global astronomical community.

1. Introduction to ALFALFA

The Arecibo Legacy Fast ALFA (ALFALFA) Survey (Giovanelli et al. 2005; Haynes et al. 2018) is a large-scale, blind extragalactic survey conducted with the 305-meter Arecibo Radio Telescope in Puerto Rico, designed to map the distribution of neutral hydrogen (HI) in the nearby Universe. Using the seven-beam Arecibo L-band Feed Array (ALFA), ALFALFA systematically scanned over 7,000 square degrees of sky between 2005 and 2012, detecting more than 30,000 HI-bearing galaxies out to a redshift of z=0.06 (Figure 1). The survey provided an unprecedented census of gas-rich galaxies, enabling studies of large-scale structure, galaxy evolution, and the cosmic HI mass function. ALFALFA's legacy lies not only in its extensive data products, but also in its foundational role connecting radio observations with multiwavelength studies of the local Universe.

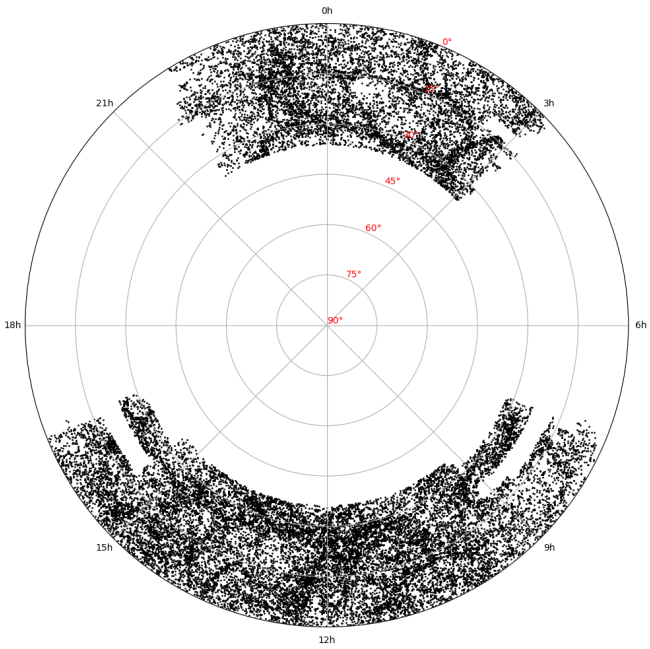


Figure 1. Plot of HI galaxy detections made with the ALFALFA Survey, centered on the North Celestial pole. Degrees of Declination lines are indicated in red and hours of Right Ascension are indicated outside the circle.

2. The NRAO Data Archive

The National Radio Astronomy Observatory (NRAO) Data Archive[1] serves as a comprehensive repository for the vast volume of observational data collected by NRAO's world-class radio telescopes, including the NSF Karl G. Jansky Very Large Array (NSF VLA), the NSF Very Long Baseline Array (NSF VLBA), and Atacama Large Millimeter/submillimeter Array (ALMA). Designed to preserve, curate, and provide open access to raw and processed radio astronomy data, the archive supports a global community of researchers exploring the Universe at radio frequencies. Through advanced data management systems, standardized metadata, and powerful search and retrieval tools, the NRAO Data Archive enables astronomers to reanalyze existing observations, perform cross-survey comparisons, and generate new scientific insights from legacy datasets. By maintaining the long-term accessibility and usability of these observations, the archive ensures that the scientific value of radio data continues to grow well beyond the lifetime of any single project.

3. ALFALFA Data

The data cubes from the Arecibo Legacy Fast ALFA (ALFALFA) Survey are three or four-dimensional data products that record the intensity of neutral hydrogen (HI) emis-

[1] `https://data.nrao.edu`

sion as a function of position on the sky and frequency (or radial velocity). Each cube typically covers several square degrees of sky, with two spatial dimensions corresponding to Right Ascension and Declination, and a third spectral dimension representing the Doppler-shifted 21-centimeter HI line. Polarization makes up the final cube dimension. These cubes were created from drift-scan observations using the Arecibo L-band Feed Array (ALFA). The data were processed to remove radio frequency interference, calibrate flux density, and correct for instrumental and atmospheric effects. The resulting data cubes allow astronomers to visualize and extract HI sources across wide areas of the sky, revealing the structure, dynamics, and distribution of gas-rich galaxies in the local Universe (Kent 2011).

We can read the original ALFALFA IDL structures into a NumPy record array. Using AstroPy (Astropy Collaboration et al. 2013) we create the set of four FITS files. The data cube (144 x 144 x 1024 x 2) metadata, and continuum metadata (144 x 144 x 2) are in each primary FITS header. The spectral cubes, spectral weights, continuum maps, and continuum weights comprise a set of sixteen cubes that can be downloaded as collections or viewed in CARTA[2].

4. Data Parameters and Download

The original ALFALFA data cubes, in proprietary IDL format, were converted with Python to FITS data cubes. Each set of sixteen FITS files contains four spectral data cubes, four spectral weights, four continuum images, and four continuum weights.

- Spectral Cubes: 144 RA pixels x 144 Decl. pixels x 1024 freq. chan. x 2 pols
- Continuum Images: 144 RA pixels x 144 Decl. pixels x 2 pols

Simply search for **A2010** at `https://data.nrao.edu/` and choose **View Images** (Figure 2). ALFALFA data cube collections with the name format **HHMM+DD** will be displayed - each collection contains 16 FITS files.

Acknowledgments. The NSF National Radio Astronomy Observatory is a facility of the U.S. National Science Foundation operated under cooperative agreement by Associated Universities, Inc.

ALFALFA observations were obtained from 2005-2012 with the Arecibo Observatory, which operated from 1963 until its untimely collapse in 2020. At the time the data were taken, the Arecibo Observatory was part of the National Astronomy and Ionosphere Center (NAIC), operated by Cornell University under a cooperative agreement with the U.S. National Science Foundation (US NSF). These data products are the work of the entire ALFALFA collaboration who have contributed to the many aspects of the survey over the years, including generations of students from the Undergraduate ALFALFA Team. Their research was supported by grants from a variety of agencies, including the US NSF and the Brinson Foundation. The ALFALFA team is indebted to - and mourns the loss of - survey lead Riccardo Giovanelli, who was an inspiration to the project.

[2] `https://cartavis.org/`

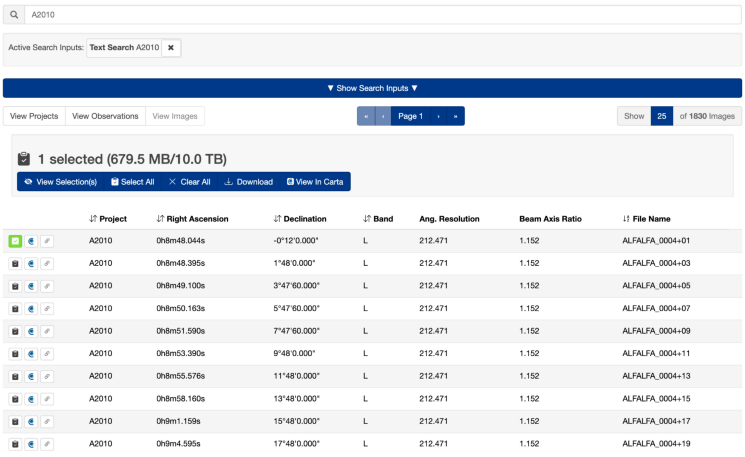


Figure 2. Downloading ALFALFA data cubes from the NRAO Data Archive.

References

Astropy Collaboration, Robitaille, T. P., Tollerud, E. J., Greenfield, P., Droettboom, M., Bray, E., Aldcroft, T., Davis, M., Ginsburg, A., Price-Whelan, A. M., Kerzendorf, W. E., Conley, A., Crighton, N., Barbary, K., Muna, D., Ferguson, H., Grollier, F., Parikh, M. M., Nair, P. H., Unther, H. M., Deil, C., Woillez, J., Conseil, S., Kramer, R., Turner, J. E. H., Singer, L., Fox, R., Weaver, B. A., Zabalza, V., Edwards, Z. I., Azalee Bostroem, K., Burke, D. J., Casey, A. R., Crawford, S. M., Dencheva, N., Ely, J., Jenness, T., Labrie, K., Lim, P. L., Pierfederici, F., Pontzen, A., Ptak, A., Refsdal, B., Servillat, M., & Streicher, O. 2013, A&A, 558, A33. eprint: 1307.6212

Giovanelli, R., Haynes, M. P., Kent, B. R., Perillat, P., Saintonge, A., Brosch, N., Catinella, B., Hoffman, G. L., Stierwalt, S., Spekkens, K., Lerner, M. S., Masters, K. L., Momjian, E., Rosenberg, J. L., Springob, C. M., Boselli, A., Charmandaris, V., Darling, J. K., Davies, J., Garcia Lambas, D., Gavazzi, G., Giovanardi, C., Hardy, E., Hunt, L. K., Iovino, A., Karachentsev, I. D., Karachentseva, V. E., Koopmann, R. A., Marinoni, C., Minchin, R., Muller, E., Putman, M., Pantoja, C., Salzer, J. J., Scodeggio, M., Skillman, E., Solanes, J. M., Valotto, C., van Driel, W., & van Zee, L. 2005, AJ, 130, 2598. eprint: astro-ph/0508301

Haynes, M. P., Giovanelli, R., Kent, B. R., Adams, E. A. K., Balonek, T. J., Craig, D. W., Fertig, D., Finn, R., Giovanardi, C., Hallenbeck, G., Hess, K. M., Hoffman, G. L., Huang, S., Jones, M. G., Koopmann, R. A., Kornreich, D. A., Leisman, L., Miller, J., Moorman, C., O'Connor, J., O'Donoghue, A., Papastergis, E., Troischt, P., Stark, D., & Xiao, L. 2018, ApJ, 861, 49. eprint: 1805.11499

Kent, B. R. 2011, in Astronomical Data Analysis Software and Systems XX, edited by I. N. Evans, A. Accomazzi, D. J. Mink, & A. H. Rots, vol. 442 of Astronomical Society of the Pacific Conference Series, 625

Astronomical Data Analysis Software and Systems XXXV
*ASP Conference Series, Vol. **VOL** Compiled: Tuesday 30th June, 2026@ 01:30*
K. Polsterer, D. Bohlender, S. Gwyn, A. Hincks and P. Teuben, eds.
*© **YEAR** Astronomical Society of the Pacific*

ZZZDEBUG!073.tex
Berriman, G, B.|textbf
Zaw, M. P, Z.
FOOBAR!conference!ADASS 2019
FOOBAR!organisations!ASP
FOOBAR! ascl:1101.010

An Interactive Metrics Dashboard for the Keck Observatory Archive

G. Bruce Berriman,[1] and Min Phone Myat Zaw,[2]

[1]*Caltech/IPAC-NExScI, Pasadena, CA 91125, USA;*
gbb@ipac.caltech.edu

[2] *University of California, 405 Hilgard Avenue, Los Angeles, CA 90095, USA*

Abstract. Since 2004, the Keck Observatory Archive (KOA) has operated as a NASA-funded collaboration between the NASA Exoplanet Science Institute (NExScI) and the W.M. Keck Observatory. It ingests and serves all data acquired by the twin 10-meter Keck telescopes on Mauna Kea, Hawaii. In the past three years, KOA has begun a modernization program to replace the architecture and systems used since the archive's creation with a new modern Python-based infrastructure. This infrastructure will position KOA to respond to the rapid growth of new and complex data sets that will be acquired by new instruments now in development, and enable follow-up to identify the deluge of alerts of transient sources expected by new survey telescopes such as the Vera C. Rubin Observatory. Since 2022, KOA has ingested new data in near-real time, generally within one minute of creation, and has made them immediately accessible to observers through a dedicated web interface. The archive is now deploying a new, scalable, Python-based, VO-compliant query infrastructure built with the Plotly-Dash framework and R-tree indices to speed-up queries by a factor of 20.

The project described here exploits the new query infrastructure to develop a dashboard that will return live metrics on the performance and growth of the archive. These metrics assess the current health of the archive and guide planning future hardware and software upgrades. This single dashboard will enable, for example, monitoring of real-time ingestion, as well as studying the long-term growth of the archive. Current methods of gathering metrics that have been in place since the archive opened will not support the archive as it continues to scale. These methods suffer from high latency, are not optimized for on-demand metrics, are scattered among various tools, and are cumbersome to use.

1. Why Build A Metrics Dashboard?

The Keck Observatory Archive (KOA) [1] has ingested all data acquired at the W. M. Keck Observatory [2] since 2004. These data become public when the period of exclusive use by the observer expires (18 months; 12 months for NASA data).

Operations are evolving rapidly to position the archive to support new science directions in astronomy. Data are now ingested in near-real time, generally within 10

[1]https://koa/ipac.caltech.edu

[2]https://keckobservatory.org/

seconds of acquisition, rather than at the end of the night. The current set of metrics-gathering tools, largely based on PHP, require an administrator to operate, create only static plots, and are generally slow. They are inadequate for gathering metrics on archive content and performance in this dynamic operations environment. This paper reports on the development of a web-based dashboard that updates metrics in near real-time every 5-7 seconds. The development is part of a long-term project to modernize the archive architecture from a C-based system to one based on a Python framework.

2. The Dashboard in Operation

The first version of the dashboard deploys the architecture and returns the number of files in the archive over time. A video demonstration of the dashboard has been posted online [3]. Figure 1 is a screenshot from this video to show the important functions of the dashboard:



Figure 1. Screenshot of the KOA metrics dashboard showing the total number of level 0 files per instrument and plots of the number of raw files for the LRIS instrument, with parameters selected from the pull-down menu.

- The data loading spinner (top right)displays when the dashboard is being updated.
- Cards showing the number of level 0 files ingested per instrument and the grand total, updated every seven seconds.
- Pull-down menus to select data for presentation: instrument, data type, date range and cadence (weekly, monthly), and year range for grand totals.
- Histograms of data in the plots, available for download in CSV format.

The remainder of this paper describes the architecture and optimization of the dashboard.

[3] https://youtu.be/QlJ-67CJNfw

3. Architectural Choices

The dashboard architecture was designed with an emphasis on widely used and maintainable Open Source technologies:

- The dashboard is integrated into the Plotly-Dash framework[4],and is the basis for the new KOA software infrastructure (Moseley et al. 2024).
- The dashboard operates with callbacks that use the Jinja 2 template engine[5] to build dynamic queries.
- The queries are submitted through the VO-compliant nexsciTAP server[6] to SQLAlchemy[7], an SQL toolkit for Python that simplifies database tasks.
- Color-coded Key Performance Indicator (KPI) cards [8]summarize archive contents.
- A backend daemon polls KOA through nexsciTAP for new metrics in near real-time and ingests them upon receipt, and records these ingestions in a log.

4. Optimizing Performance for Updates at a Seven-Second Cadence

A key part of the design is to the dashboard is able to support the cadence at which data are ingested.

4.1. Database Design

Database tables have been created for each year of observatory operations. The schema has been designed to minimize data transfer from the DBMS to the dashboard:

- The time and instrument values are the primary columns.
- The numbers of science and calibration files are composed as virtual columns.
- SQL templates create simplified queries.

4.2. Parallel Queries in the Backend Systems Architecture

Pre-calculated, static database tables from 1994-2024 have been created on a daily, monthly, and annual basis. The tables for the current year, 2025 at the time of the presentation, are dynamic. Parallel queries to these two sets of tables improve performance, as shown in Figure 2.

Performance is continuously monitored by the ingestion daemon, and by the Plotly built-in event monitoring system, which graphically displays the update speeds in milliseconds.

---

[4] `https://dash.plotly.com/`

[5] `https://jinja.palletsprojects.com/en/stable/`

[6] `https://github.com/Caltech-IPAC/nexsciTAP`

[7] `https://www.sqlalchemy.org/`

[8] `https://www.kpi.org/`

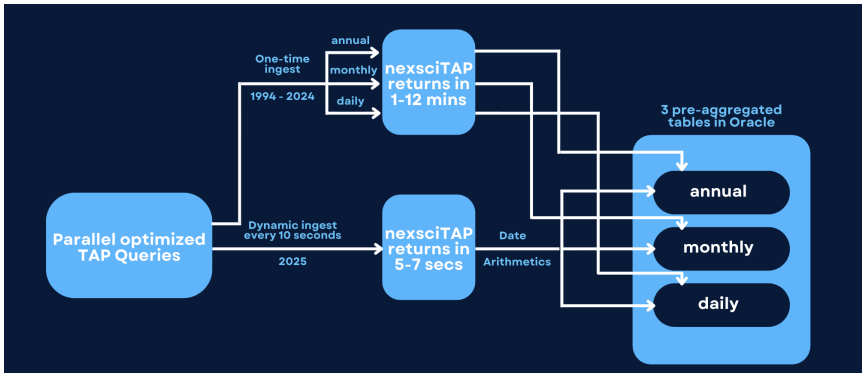


Figure 2. Block diagram of the use of parallel queries to the database that improve performance.

5. Conclusions

The dashboard architecture presented here meets the archive's goal of monitoring the number of files ingested by the archive during a calendar year in near real-time. In the coming year, we will expand the dashboard to include a comprehensive set of metrics, such as data volume, the number of archive queries, and download volumes. We will also augment functionality as needed, such as overlay plots for selected instruments.

Acknowledgments. The Keck Observatory Archive (KOA) is a collaboration between the NASA Exoplanet Science Institute (NExScI) and the W. M. Keck Observatory (WMKO). NExScI is sponsored by NASA's Exoplanet Exploration Program and operated by the California Institute of Technology in coordination with the Jet Propulsion Laboratory (JPL). Mr. Zaw developed this dashboard as a 2025 Caltech Summer Undergraduate Research Fellow (SURF). We thank the KOA team for supporting this project, and we thank Ms. Marcy Harbut for proof reading the paper.

The observatory was made possible by the generous financial support of the W. M. Keck Foundation. The authors wish to recognize and acknowledge the very significant cultural role and reverence that the summit of Mauna Kea has always had within the indigenous Hawaiian community. We are most fortunate to have the opportunity to conduct observations from this mountain.

References

Moseley, R., Berriman, G. B., Gelino, C. R., Good, J. C., & Oluyide, T. 2024, in ADASS XXXIV, edited by A. DeMarco, & J. Said (San Francisco: ASP), vol. TBD of ASP Conf. Ser., 999 TBD

Astronomical Data Analysis Software and Systems XXXV
*ASP Conference Series, Vol. **VOL** Compiled: Tuesday 30th June, 2026@ 01:30*
K. Polsterer, D. Bohlender, S. Gwyn, A. Hincks and P. Teuben, eds.
*© **YEAR** Astronomical Society of the Pacific*

ZZZDEBUG!076.tex
Coulson, J.|textbf
Miszalski, B.
O'Toole, S. J.
Lidman, C.
Onken, C.
Tocknell, J.
Sheng, K.
web!Apache NiFi
archives!design
archives!Multiple!Data
Central Archives
archives!reduced
data!pipelines
data!processing!automated
instruments!individual!WiI
PyWiFeS! ascl:1402.034

Automating Data Transfer, Archival Workflows and Data Reduction Pipeline Execution with Apache NiFi

James Coulson,[1] Brent Miszalski,[1] Simon J. O'Toole,[1] Christopher Lidman,[2] Christopher Onken,[2] James Tocknell,[1] and Kate Sheng[1]

[1]*Australian Astronomical Optics - Macquarie, Faculty of Science and Engineering, Macquarie University, North Ryde, New South Wales, Australia;* james.coulson@mq.edu.au

[2]*Research School of Astronomy and Astrophysics, The Australian National University, Canberra, Australian Capital Territory, Australia*

Abstract. Most traditional data reduction pipelines are on run an investigators local machines or remote machines requiring a manual touch to be executed. This approach leads to data discoverability and reproducibility issues. Additionally, observatory sites are also often remote and one of the major challenges is facilitating data transfer from site to site. Data Central's Apache NiFi system aims to combine data transfer, archival workflows and triggering pipeline execution to produce discoverable data products while managing the complexity of secure site to site data transfer. We cover the implementation and design of this system for Data Central's archive of the ANU 2.3m Telescope at Siding Spring Observatory and how we have designed our system to enable scalable data transfers, data transformation and pipeline execution for incoming observations. We also highlight how archival processes are integrated to improve data discoverability ensuring data products are easily located, shared and reused. We discuss possible future directions including automatic notification systems (such as for transient events) and multi instrument or multi observatory abilities.

1. Introduction

Traditionally, data reduction pipelines are often manually executed on investigators local machines or on single remote compute servers. This approach requires significant effort to configure, validate and maintain environments, diverting time from producing scientific results. Although functional, it can be difficult to manage these workflows at scale and can be a challenge to reproduce. As data volumes have grown substantially, long-term preservation and discoverability of raw data and reduced data products have remained a challenge.

Throughout the data lifecycle, additional challenges arise particularly in the ingestion of observations into archival systems. Many observatories are situated in remote locations with limited network connectivity, and data transfer pipelines are built using inconsistent tooling with manual, labour-intensive processes. These factors contribute to limited operational observability and monitoring capabilities, increasing operational overhead and the likelihood of transfer failures or data loss.

The ANU 2.3m Telescope at Siding Spring Observatory (SSO) in north-western New South Wales has been in automatic operation since March 2023 (Price et al. 2024). A critical part of this operational model is an automated end-to-end archival system to allow rapid transient follow up. This requires reliably transferring data from the observatory and making it discoverable shortly after observations have been completed. The designed workflow handles three stages as part of this model: site-to-site data transfer, automated data reduction pipeline execution and executing archival workflows. To implement this, we propose a system orchestrated using Apache NiFi as a flexible scalable solution to the challenges presented.

2. Apache NiFi

Apache NiFi[1] is an open source software project maintained by Apache. The primary purpose of NiFi is to implement an Extract-Transform-Load (ETL) data transformation model for large volumes of unstructured data from multiple sources. NiFi has been widely implemented in several other industries such as manufacturing, finance, logistics, and agriculture to manage large continuous streams of unstructured data that require reliable transfer, transformation, and observability. A web-based graphical interface provides a low-code, click and drag environment developers can use to design, control and monitor data flows.

NiFi implements a "flow-based" approach to data. The main advantage offered by this type of design is guaranteed data delivery. The queue system has been designed to be fault-tolerant by holding files in case of system failures, while being back pressure controlled in case of performance bottlenecks. Data can also be routed based on the result or status of a processor. For example, failed files can be placed into a holding directory for re-ingestion and record the failures for debugging.

There are several processors built into NiFi that perform the transform stage of the workflow. These processors perform specific actions on files throughout the workflow to clean, validate and enrich data. Developers can use the framework provided to create custom processors or custom Python, Java or Groovy scripts.

Once the files are parsed, we then need to load them into the archive database and into storage to archive. NiFi provides connectors for a wide range of database systems for both local backends (such as MongoDB) and cloud based backends (such as Azure and S3).

3. System Design

3.1. File Transfer and Ingestion

With Apache NiFi as the backbone of the system, we have designed integrations with the automated data reduction system and the Data Central (Mannering & O'Toole 2019) Archive. Figure 1 shows a high level overview of the system design from observations at the telescope through to the data being available through the archive frontend. While NiFi is the central orchestrator of the system, there are various other technologies integrated to complete the full end-to-end workflow.

[1] https://nifi.apache.org/

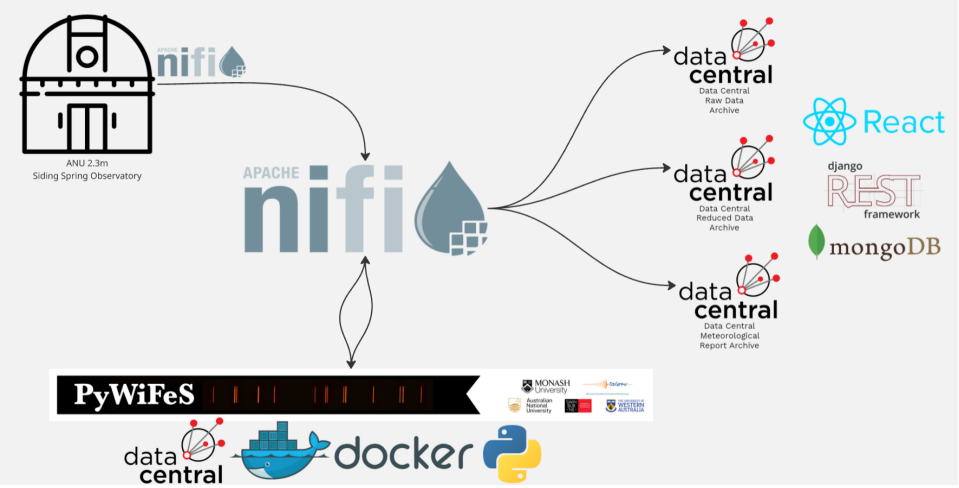


Figure 1. Data Central Archive System Design

The first stage is the site-to-site data transfer. The telescope itself runs its own instance of NiFi that watches a directory for any new files. This means that files are transferred immediately allowing access shortly to raw files shortly after observations are taken. The data transfer takes place over an HTTPS connection secured with TLS using NiFi's inbuilt tooling to Data Central's own NiFi instance.

Once received, NiFi uses our distributed storage system as the backend storage (Ceph)[2] and written to disk. A python script is then executed to extract relevant metadata such as FITS headers and ObsCore (Louys et al. 2017) fields to create an entry in the MongoDB database. Any errors processing the files are logged to create an end of night report.

A separate workflow to ingest meteorological data has also been implemented. This workflow ingests each day's weather report and investigators can query for them inside the archive front end to be served as an HTML webpage.

3.2. Pipeline Execution

Automated pipeline execution is handled by a bespoke Dockerised pipeline orchestration system written in Python. It runs the PyWiFeS 2.0 data reduction pipeline which has been upgraded to utilise multiprocessing resulting in performance improvements and the ability to run automated reductions.

The reduction system continuously queries the archive database for new or existing observations for reduction. Symlinks are then automatically gathered for science and nearest calibration frames to avoid data duplication. The data is then reduced to generate the data products and a pipeline report is produced. The pipeline report is a JSON formatted document which is used to keep track of failures and other provenance data such as success, weather, input files and execution time. Re-reductions are triggered if the pipeline has failed or if new data is available such as science frames, skyflats or standards.

[2] https://ceph.io/en/

NiFi then watches the reductions directory for any new reductions and ingests the pipeline report into the archive database.

4. Data Central Archive

Data Central provides an archive of both raw and reduced data built using React and Django Rest Framework. Users can query and download data through the web interface. Users can also download either data products only or include input files when downloading reduced data allowing an easy way to verify results or manually run a re-reduction. Access control is built into the archive, with science frames subject to an 18 month proprietary period. Investigators can set access through LENS, the proposal portal. LENS has been integrated to give anyone listed as an investigator on the proposal access automatically as data becomes available and access can be granted to other users without listing them as an investigator or an observer.

5. Future Directions

To extend the functionality of the system, integrations with alert brokers such as Fink (Möller et al. 2021) could be considered to automate execution of transient event observations and analysis by consuming alert streams provided for additional functionality. There is futher potential to automate Virutal Observatory protocols such as TAP to provide real time updates to metadata.

NiFi is a strong candidate to act as the data backbone across multiple facilities and instruments as it provides a standardised workflow with built-in data stream aggregation tools. The extensive list of processors and the ability to extend functions allow the ability to implement more bespoke workflows and processing as required. For example, there are future plans to adapt the system described here to reduce data from the Veloce instrument on the Anglo Australian Telescope at SSO.

Improvements to telescope operations can also be improved as NiFi provides real time observability and can be paired with a monitoring dashboard such as Grafana for more in depth insights into performance.

References

Louys, M., Tody, D., Dowler, P., Durand, D., Michel, L., Bonnarel, F., Micol, A., & IVOA DataModel Working Group 2017, Observation Data Model Core Components, its Implementation in the Table Access Protocol Version 1.1, IVOA Recommendation 09 May 2017

Mannering, E., & O'Toole, S. 2019, in Linking Galaxies from the Epoch of Initial Star Formation to Today, 11

Möller, A., Peloton, J., Ishida, E. E. O., Arnault, C., Bachelet, E., Blaineau, T., Boutigny, D., Chauhan, A., Gangler, E., Hernandez, F., Hrivnac, J., Leoni, M., Leroy, N., Moniez, M., Pateyron, S., Ramparison, A., Turpin, D., Ansari, R., Allam, T., Jr., Bajat, A., Biswas, B., Boucaud, A., Bregeon, J., Campagne, J.-E., Cohen-Tanugi, J., Coleiro, A., Dornic, D., Fouchez, D., Godet, O., Gris, P., Karpov, S., Nebot Gomez-Moran, A., Neveu, J., Plaszczynski, S., Savchenko, V., & Webb, N. 2021, MNRAS, 501, 3272. eprint: 2009.10185

Price, I., Nielsen, J., Lidman, C., Soon, J., Travouillon, T., & Sharp, R. 2024, Publications of the Astronomical Society of Australia, 41, e057. eprint: 2409.19842

Astronomical Data Analysis Software and Systems XXXV
*ASP Conference Series, Vol. **VOL** Compiled: Tuesday 30th June, 2026@ 01:30*
K. Polsterer, D. Bohlender, S. Gwyn, A. Hincks and P. Teuben, eds.
*© **YEAR** Astronomical Society of the Pacific*

ZZZDEBUG!079.tex
Ali, E.|textbf
Serene, A.
Becker, V.R.
Aranda, S.I.
Kleinman, S.J.

Designing the Nightly Digest by ADACS for Rubin Observatory

Eman Ali[1], Alice Serene[1], Valerie R. Becker[2], Sebastian I. Aranda[3], and Scot J. Kleinman[4,5]

[1]*Swinburne University of Technology, Melbourne, VIC, Australia;* eali@swin.edu.au

[2]*Vera C. Rubin Observatory/NSF NOIRLab, Tucson, AZ, USA*

[3]*Vera C. Rubin Observatory/NSF NOIRLab, La Serena, Coquimbo, Chile*

[4]*AURA, Rubin Observatory Construction, Tucson, AZ, USA*

[5]*Astromanager LLC, Hilo, HI, USA*

Abstract. The Nightly Digest (ND) is a web-based dashboard developed collaboratively by the Astronomy Data and Computing Services (ADACS) team and the Vera C. Rubin Observatory Telescope and Site Software team to summarize nightly observatory operations. It condenses complex operational logs and telemetry into a small set of visual indicators and key metrics supporting situational awareness, decision-making, and handovers for a diverse group of stakeholders. This paper presents the design process and lessons learned in creating the ND and highlights the role of iterative, user-centered, and collaborative design in astronomy operations software.

1. Project Context

The NSF-DOE Vera C. Rubin Observatory entered Operations at the end of 2025 and is preparing to commence the Legacy Survey of Space and Time (LSST, Ivezić et al. 2019).[1] The survey will image the entire southern sky twice per week over the next decade, generating many terabytes of data each night. Alongside science data products, observatory systems produce a large volume of operational logs, metrics, and event records.

A recurring operational question, asked by night staff, scientists, and management alike, is deceptively simple: how did the night actually go? Answering this question requires synthesizing heterogeneous data sources into a coherent and accessible summary that supports situational awareness and efficient handovers among a broad range of users.

During construction, Rubin teams developed a number of tools that target specific operational needs; however, there was no single comprehensive view that brought these perspectives together. This gap motivated the development of the Nightly Digest (ND),

[1]https://noirlab.edu/public/images/Revelado-Digital-Aura-LasTacas-N-46-CC/

tware!design
ernational
llaboration
ervatories!ground-based!Rubin
bservatory
veys!LSST
tware!logging
!application

a web application designed to condense nightly observatory operations into a clear set of visual indicators and key performance metrics.

The ND originated as an interactive Jupyter notebook developed by the Rubin Logging team to explore available data sources. Deployed as a Times Square (Sick 2021) notebook on the Rubin Science Platform (RSP, O'Mullane et al. 2024), this early prototype enabled open access and rapid feedback from managers and scientists. Through a series of workshops, exploratory ideas were refined into use cases and formal requirements. The Astronomy Data and Computing Services (ADACS) team subsequently joined the project, leading the design effort to transform these requirements into a clean, flexible, and production-ready dashboard.

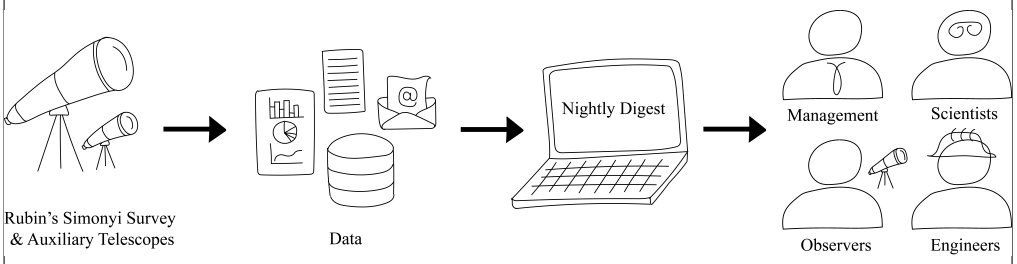


Figure 1. Operational context of the Nightly Digest within Rubin Observatory nightly operations.

2. Lessons Learned

During the design and development of the ND, our team faced a range of challenges that led to a number of important lessons. The following eight lessons capture the most significant insights from this work.

2.1. Solve the Real Problem

What users say they want is not necessarily what they need. Use cases and requirements provide valuable guidance, but delivering a tool that satisfies documented requirements without addressing the underlying problem offers limited value.

To understand the real problem, direct observation of users is essential. A member of our team visited the summit, observed operational workflows, and recorded detailed notes. Where multiple user groups are involved, observing each in context helps reveal differing needs, constraints, and priorities.

2.2. Draw First

Once the problem space is understood, early design work benefits from low-fidelity sketching. At this stage, pen and paper are more effective than digital tools. Sketching ideas encourages exploration without premature commitment.

Early concepts are rarely the best solutions. Producing and discarding incorrect ideas helps clarify better ones and accelerates progress toward more effective designs.

software!design!user requirements
techniques!dashboards
techniques!data challenges
software!user interfaces
software!user interfaces!tooltips

2.3. Start Small, Evolve Fast

At the outset, the project included more than twenty use cases and forty requirements, which was initially overwhelming. Progress was achieved by focusing on one element at a time: a single requirement, metric, data representation, or user interface element.

Design is an iterative process. Starting with low-fidelity wireframes and incrementally refining them into high-fidelity designs allowed the system to evolve in manageable steps.

2.4. Show, Don't Just Tell

Shared terminology does not guarantee shared understanding, particularly when discussing systems that do not yet exist. We repeatedly encountered situations where team members used the same words but referred to different concepts.

Sharing sketches and mockups proved essential for aligning understanding. Encouraging others to contribute their own drawings helped surface differing interpretations of requirements. When requesting feedback, specific questions were more effective than general ones; for example, asking whether information was missing, labels were clear, or defaults were appropriate.

Design remained iterative throughout this process: draw, share, gather feedback, and repeat.

2.5. Keep Users in the Loop

Once implementation began, it became evident that users needed meaningful feedback about the state of the Nightly Digest application itself, including data availability, software failures, and delivery progress. The ND provides actionable error messages with paths to restore normal ND operation.

Delivery expectations are managed through placeholders and estimated timelines for incomplete features, alongside alternative ways to access required information. Contextual tooltips document data sources, calculations, and terminology. Keeping users informed builds trust and makes errors easier to understand and manage.

2.6. Ship it, then Sharpen it

The ND was released incrementally, prioritizing speed over completeness or visual polish. The initial version focused on delivering the most critical elements, particularly the summary metric cards, while deferring less essential features. This approach allowed users to gain value early and enabled the team to gather feedback that informed subsequent refinements. Over time, the system was iteratively improved based on real usage rather than assumed requirements.

2.7. Design for the Edges

Although the ND primarily answers how a single night went, several use cases required working across multiple nights. This introduced edge cases such as marking night boundaries in plots, navigating long textual reports, and dynamically showing or hiding visit information across nights in visualizations. Usability across screen sizes introduced further constraints, requiring responsive layouts and removal of secondary elements on smaller displays to prioritize data visibility.

2.8. More Signal, Less Noise

The ND is highly information-dense, making careful management of visual hierarchy essential; the most important information is presented prominently, while secondary details remain accessible through interactions such as hovering, clicking, or using dropdowns. Default views are tailored to the average user, while advanced configuration menus are accessible for our expert users. Interaction patterns such as synchronized zooming across plots further reduce complexity while supporting deeper exploration when required.

3. Summary and Guidance

These lessons reflect the iterative, observational, and collaborative approach that guided the Nightly Digest from an exploratory prototype to a production-ready dashboard, supporting diverse stakeholders while maintaining clarity and usability.

3.1. Additional Guidance

For teams seeking practical guidance on user experience (UX) design, NN/g (The Nielsen Norman Group) is a recommended resource.[2] NN/g draws on over two decades of human–computer interaction research and provides extensive free materials and professional training. Their evidence-based approach supports informed design decisions, particularly for iterative, user-driven projects.

Acknowledgments. This material is based upon work supported in part by the National Science Foundation through Cooperative Agreements AST-1258333 and AST-2241526 and Cooperative Support Agreements AST-1202910 and 2211468 managed by the Association of Universities for Research in Astronomy (AURA), and the Department of Energy under Contract No. DE-AC02-76SF00515 with the SLAC National Accelerator Laboratory managed by Stanford University. Additional Rubin Observatory funding comes from private donations, grants to universities, and in-kind support from LSST-DA Institutional Members.

ADACS is funded from the Astronomy National Collaborative Research Infrastructure Strategy (NCRIS) allocation provided by the Australian Government and managed by Astronomy Australia Limited (AAL).

References

Ivezić, Ž., et al. 2019, ApJ, 873, 111. eprint: 0805.2366, URL https://doi.org/10.3847/1538-4357/ab042c

O'Mullane, W., et al. 2024, in Astromical Data Analysis Software and Systems XXXI, edited by B. V. Hugo, R. Van Rooyen, & O. M. Smirnov, vol. 535 of Astronomical Society of the Pacific Conference Series, 227. eprint: 2111.15030, URL https://doi.org/10.48550/arXiv.2111.15030

Sick, J. 2021, The Times Square service for publishing parameterized Jupyter Notebooks in the Rubin Science platform, SQuaRE Technical Note SQR-062, NSF-DOE Vera C. Rubin Observatory. URL https://sqr-062.lsst.io/

[2] https://www.nngroup.com/

Astronomical Data Analysis Software and Systems XXXV
*ASP Conference Series, Vol. **VOL** Compiled: Tuesday 30th June, 2026@ 01:30*
K. Polsterer, D. Bohlender, S. Gwyn, A. Hincks and P. Teuben, eds.
*© **YEAR** Astronomical Society of the Pacific*

ZZZDEBUG!081.tex
Moseley, R.|textbf
Berriman, G. B.
Gelino, C. R.
Good, J. C.
Good, J. C.
Good, J. C.
Oluyide, T.
Moseley, R.
Berriman, G. B.
Gelino, C. R.
Good, J. C.
Lynn, M.
Swain, M.
Oluyide, T.
Montage! ascl:1010.036

A New Plotly-Dash-based Query Infrastructure for the Keck Observatory Archive

R. Moseley,[1] G. Bruce Berriman,[1] Christopher R. Gelino,[1] John C. Good,[1] Meca Lynn,[1], Melanie Swain,[1] and Toba Oluyide[1]

[1]*Caltech/IPAC-NExScI, Pasadena, CA 91125, USA;*
moseley@ipac.caltech.edu

Abstract. The Keck Observatory Archive (KOA) curates all observational data acquired at the W. M. Keck Observatory. The archive is expected to grow rapidly as complex new instruments are commissioned and as the expectations of archive users have expanded. In response, KOA has implemented a new Python-based, VO-compliant query infrastructure. This work is a continuation of the architectural design and technology selection identified at ADASS 2024. We have deployed real-time ingestion of newly acquired data and a dedicated interface for observers to manage these data. Our ADASS 2024 poster identified the new technologies chosen: Plotly-Dash, a low-code framework that exploits event-driven callbacks to simplify the handling of user interactions; R-tree spatial indexing to speed up spatial searches by ×20; a VO-compliant TAP middleware, already in use at the NASA Exoplanet Archive and NEID archive; and `mViewer`, a visualization engine in the Montage Image Mosaic toolkit that is optimized for astronomy images.

These technologies will underpin new services that can be hosted on web pages or in Jupyter notebooks, and when completed, will replace the current query infrastructure. We have completed two new services now in beta release. The first is the Data Discovery Service, a web-based dashboard that returns spatial and temporal queries of the entire archive in seconds. It supports filtering observations by keywords, previewing results in an interactive data grid, and visualizing images, and it offers data downloads. The second is a Jupyter notebook that performs interactive visualization of Keck observations of protostars in the Rho Oph Dark Cloud and uses data from CDS and IRSA, as well as KOA.

1. Introduction

The Keck Observatory Archive (KOA) manages over 30 years of raw and reduced data, from all current and decommissioned W. M. Keck Observatory instruments. The archive is currently modernizing its operations and query infrastructure to respond to changing directions in research and to serve massive datasets from the next generation of instruments. Modernization prioritizes widely adopted, open-source, and sustainable Python technologies wherever possible. This paper describes two services that use the core technologies, described in Moseley et al. (2024) and summarized below, which have been demonstrated in the following two YouTube videos:

- A Data Discovery Service (DDS) that queries the archive in seconds[1]
- A Jupyter notebook that uses the DDS to discover Keck data and integrate them with remote datasets, accessed through Virtual Observatory (VO) protocols[2].

2. Data Discovery Service

The Data Discovery Service (DDS) is a major part of the modernization of the KOA query infrastructure. The DDS takes a "low-code," Python-centric approach to architecture through the Plotly-Dash framework. This framework uses event-driven callbacks to manage complex client-server interactions, including coordinate resolution and quick spatial searches, without the overhead of maintaining a decoupled JavaScript front end. Figure 1 shows a block diagram of the DDS functionality.

The video demonstration performs a spatial search of a 6°radius centered on M51, which has been observed with every instrument at the Keck Observatory since it began operating in 1994. The DDS queried the entire archive of six million records in approximately 15 seconds. The return is shown in Figure 2. The left column shows an instrument-by-instrument summary; on the right is an interactive tabbed display of the metadata.

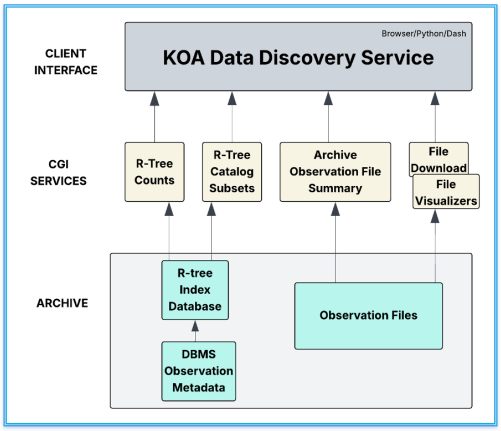


Figure 1. Overview of the Data Discovery Service (DDS) functional design

The DDS uses the following technologies:

- `nexsciTAP`, a Python Table Access Protocol (TAP) implementation that provides a VO-compliant mechanism for handling Astronomy Data Query Language (ADQL) requests.
- R-tree spatio-temporal indices that approximate complex instrument geometries using minimum bounding rectangles (MBRs). Grouping nearby objects into hi-

[1] `https://www.youtube.com/watch?v=RMtqsMjeh-Y`

[2] `https://www.youtube.com/watch?v=UC6-tuWWYQo`

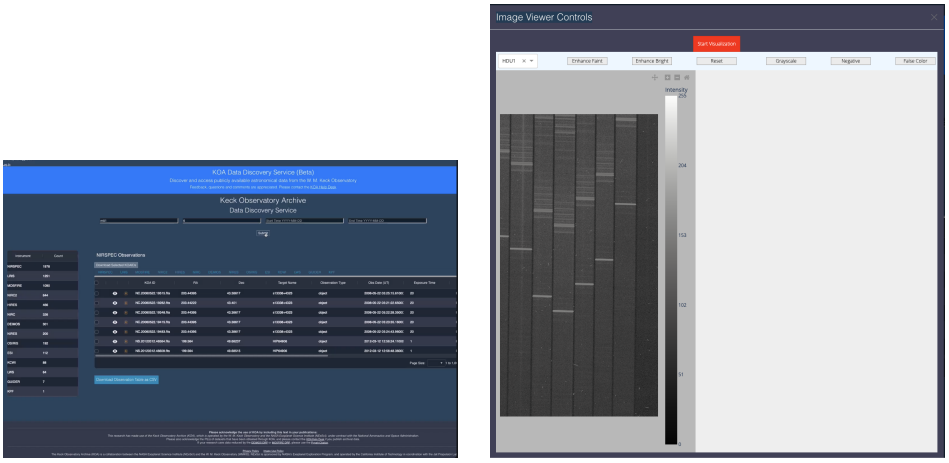


Figure 2. The Data Discovery Service (DDS) return page. *Left:* The main search and results dashboard showing the instrument-based observation summary and filtered data table *Right:* The integrated `mViewer` component displaying a DEIMOS FITS image preview with interactive histogram and stretch controls

erarchical MBRs (Guttman 1984) instantiated as memory-mapped files[3]. R-tree indices allow fast searches, ×20 faster than current services.

- A Dash React component wraps the `mViewer` module of the Montage Image Mosaic Engine (Berriman & Good 2017) to provide an adaptive, histogram-based visualizer.

- `wget` shell scripts for bulk data retrieval.

3. KOA DDS Notebook

We have delivered an example Jupyter notebook that uses the DDS to interact with remote observations of Rho Ophiuchi (Rho Oph). Two screenshots from the video[4] are shown in Figure 1. This second demonstration details the following workflow: after resolving the coordinates of Rho Oph with SIMBAD and defining a search radius, the DDS discovers all Keck observations in Rho Oph, retrieves a catalog of protostars in Rho Oph from VizieR (Lindberg et al. 2017), and overlays these data on a Spitzer IRAC mosaic acquired from IRSA

4. Conclusion

The Plotly-Dash framework provides a powerful mechanism for building a new query infrastructure for KOA. The DDS and the associated notebook are examples of powerful

[3] `https://github.com/Caltech-IPAC/Montage`

[4] `https://www.youtube.com/watch?v=UC6-tuWWYQo`

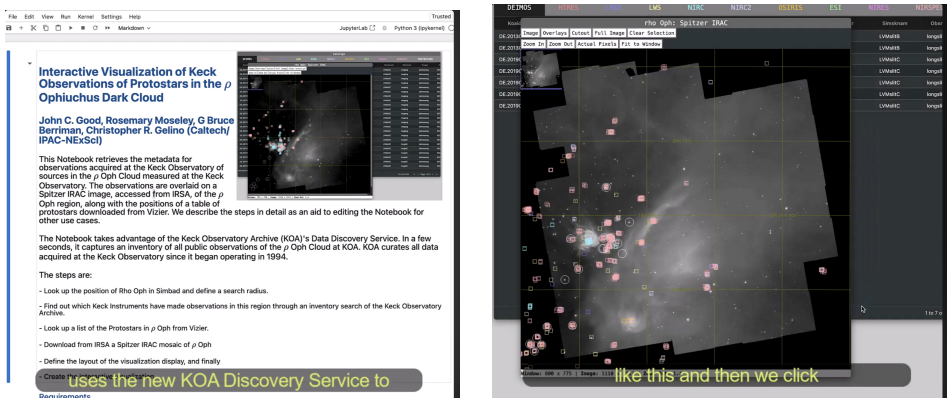


Figure 3. Jupyter Notebook integration with the DDS API. *Left:* Notebook usage of Keck observations and external VO catalogs within a Jupyter environment *Right:* The resulting interactive visualization overlaid on the Spitzer IRAC mosaic

new services that can be built with the new query infrastructure, and will deliver new tools to serve the needs of the rapidly expanding archive.

Acknowledgments. The Keck Observatory Archive (KOA) is a collaboration between the NASA Exoplanet Science Institute (NExScI) and the W. M. Keck Observatory (WMKO). NExScI is sponsored by the NASA Exoplanet Exploration Program and operated by the California Institute of Technology in coordination with the Jet Propulsion Laboratory (JPL).

The observatory was made possible by the generous financial support of the W. M. Keck Foundation. The authors wish to recognize and acknowledge the very significant cultural role and reverence that the summit of Mauna Kea has always had within the indigenous Hawaiian community. We are most fortunate to have the opportunity to conduct observations from this mountain.

References

Berriman, G. B., & Good, J. C. 2017, PASP, 129, 058006. eprint: 1702.02593

Guttman, A. 1984, in Proceedings of the 1984 ACM SIGMOD International Conference on Management of Data (New York, NY, USA: Association for Computing Machinery), SIGMOD '84, 47. URL https://doi.org/10.1145/602259.602266

Lindberg, J. E., Charnley, S. B., Jørgensen, J. K., Cordiner, M. A., & Bjerkeli, P. 2017, ApJ, 835, 3. eprint: 1611.06976

Moseley, R., Berriman, G. B., Gelino, C. R., Good, J. C., & Oluyide, T. 2024, in ADASS XXXIV, edited by A. DeMarco, & J. Said (San Francisco: ASP), vol. TBD of ASP Conf. Ser., 999 TBD. eprint: 2412.12356, URL https://arxiv.org/abs/2412.12356

Astronomical Data Analysis Software and Systems XXXV
*ASP Conference Series, Vol. **VOL** Compiled: Tuesday 30th June, 2026@ 01:30*
K. Polsterer, D. Bohlender, S. Gwyn, A. Hincks and P. Teuben, eds.
*© **YEAR** Astronomical Society of the Pacific*

ZZZDEBUG!082.tex
Huynh, S.|textbf
Tocknell, J.
computer
languages!Python
software!performance
databases!Postgres
software!tools!Django
data!ingestion

Performance Profiling and Monitoring of the Data Central Ingestion System

Sam Huynh, and James Tocknell

Australian Astronomical Optics, Macquarie University, Sydney, NSW, Australia

Abstract. As astronomy data sets become larger, efficient data ingestion systems into platforms such as Data Central are required to ensure science ready data products are promptly available to the community. Within Data Central's ingestion system, bottlenecks have been identified with the use of `py-spy`[1] and diagnostic queries against the ingestion database. Rectifying inefficient database usage resulted in an 8 times speed up of some ingestion components. This paper discusses the identification and addressing of these bottlenecks, some useful features and tricks to be aware of when using `py-spy`, and the importance of robust logging and performance monitoring in long running systems.

1. Introduction

Data Central[2], Australia's astronomy research platform, (Mannering & O'Toole 2019) contains an ingestion system to serve data products from survey teams in a FAIR manner.[3] The ingestion system processes data products provided by survey teams to database records, allowing users to access the survey's data products through a web interface or API. Data Central's ingestion system consists of the following:

- a Django backend
- a PostgreSQL database for Django
- a PostgreSQL database cluster for TAP
- multiple Celery workers

The execution of these components is orchestrated through Docker Compose to create reproducible environments where each component can communicate with each other. The Django backend is responsible for processing each data product and its metadata (such as catalogue column descriptions and spectra documentation) which are encapsulated in tasks sent to Celery workers to take advantage of parallelisation. Data products directly supported by Data Central include catalogues, imaging, spectra and integral-field spectroscopy products. Other data products (such as PDFs) are ingested as blobs.

[1] `https://github.com/benfred/py-spy`

[2] `https://datacentral.org.au`

[3] `https://www.go-fair.org/fair-principles/`

2. Identifying the problem

Data releases take place two to three time each year based on community requests and involve data volumes of up to 20 TBs. In preparation for a given data release, Data Central will run the ingestion system over subsets of the data release to ensure data integrity and to check that Data Central's user interfaces correctly serve ingested data products before ingesting the full data release. For GALAH DR4, the less-used, and thus immature, blob ingestion component was required to ingest additional data products provided by the survey team.

Initial ingestion runs provided estimates for the time required to run the ingestion system on the full data release. The blob ingestion component was estimated to require 8 days of run time to complete. In contrast, the remaining ingestion system components was estimated to require 2 days of run time.

The blob ingestion component consists of querying the PostgreSQL database for metadata for correct references, file size calculation and writing new metadata into the database. These operations are simpler than spectral data ingestion which involves validating that spectral data files conform to FITS standards and provide sensible metadata within header keywords.

2.1. Monitoring PostgreSQL activity

Database systems usually have a mechanism to monitor query activity and performance statistics built into the system. PostgreSQL provides the `pg_stat_activity`[4] view which provides details about each server process within PostgreSQL.

The following diagnostic checks which queries are running on the PostgreSQL database. This includes the time spent executing the query.

```
SELECT pid,
age(clock_timestamp(), query_start),
usename, query
FROM pg_stat_activity
WHERE query != '<IDLE>'
AND query not ILIKE '%pg_stat_activity%'
ORDER BY query_start desc;
```

This diagnostic query is useful in determining what load the PostgreSQL database is experiencing. The following snippet shows an example of a long running query - an explicit database lock that has not been released.

```
pid   |       age       | usename  |           query
--------+-----------------+----------+---------------------------
135992 | 00:04:19.081318 | postgres | LOCK t;
```

The following diagnostic checks if any query is blocked by another query and prints the query and the PID of the blocker.

```
SELECT pid, usename, pg_blocking_pids(pid) as blocked_by,
query as blocked_query
FROM pg_stat_activity
WHERE cardinality(pg_blocking_pids(pid)) > 0;
```

[4] https://www.postgresql.org/docs/current/monitoring-stats.html#MONITORING-PG-STAT-ACTIVITY-VIEW

This diagnostic query is useful to determine if a single (or singular type) of query is blocking queries. The following snippet shows an `INSERT` query is blocked by the previous table lock.

```
pid    | usename  | blocked_by |       blocked_query
--------+----------+------------+---------------------------
140895 | postgres | {135992}   | INSERT INTO t VALUES (3);
```

Monitoring both queries over time clarifies what queries are causing a bottleneck in the system. For example, a write query may block multiple write queries on the same record. The first query would print all write queries that are running and the second query will show which write queries are blocked by a single query.

In the case of the blob ingestion component for Data Central, the expected queries appearing in the diagnostic query result are write queries due to record creation for each data product. Many read queries were in the diagnostic query result. Additionally, many of the read queries were repeated. This gave evidence for a suboptimal query execution pattern within the blob ingestion component.

The extension `pg_stat_statements` is often useful for providing statistics on query planning and execution. In this instance, however, no useful information was obtained from `pg_stat_statements` that was not provided by the two queries on `pg_stat_activity`.

2.2. py-spy

The Python package `py-spy` is a profiler which visualises the frequency of code line execution within a Python program by sampling Python's call stack. This is done without making any modifications to the program. After profiling a Python program with `py-spy record -- python <program>`, a flame graph[5] is produced where the call stack samples are broken down to each function call and aggregated (Gregg 2016).

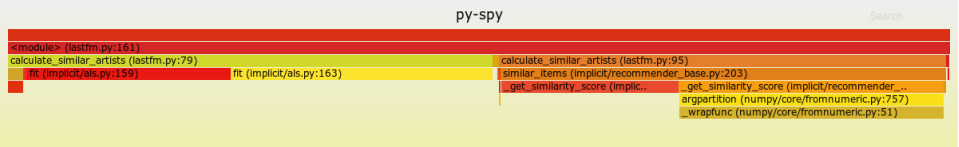


Figure 1. An example flame graph. The third row shows two lines within a function, each taking approximately half of the program's execution time. The subsequent rows further divide the execution time with calls to other functions.

Due to the ingestion system running through Docker Compose, additional capabilities are required to enable `py-spy` to read Python's call stack. The capability `SYS_PTRACE` enables `py-spy` to function within Docker containers and Kubernetes pods. Additionally, database queries are run directly from the main ingestion system and the Celery worker. Thus, `py-spy` will be used in both containers to obtain a complete view of the ingestion system when running blob ingestions. `py-spy` provides a mode to monitor Python programs already running.

[5] `https://github.com/brendangregg/FlameGraph`

Upon profiling the Django program and Celery worker, it was found that time was wasted by repeating the same read queries from the database, and inefficient parallelisation caused by the large number of queries writing single records.

3. Fixing the problem

To rectify the wasted time repeating read queries, the Django program was modified to process blob data products in batches rather than isolated tasks. This allowed results to be shared for blobs relating to the same object. Additionally, the batching of blob data products enabled the batching of records in the write query processed by the Celery worker. After tuning the batch size, the total run time for the blob ingestion component was reduced from 8 days to less than 1 day.

4. Long term maintenance

Due to the infrequent use of the blob ingestion component compared to other ingestion components, appropriate logging messages were lacking and appropriate metrics such as database record write rate were missing. These can provide additional context to target the diagnostic process. Logging messages were added to accurately reflect the state of the ingestion system. Log collection to a central system would enhance the usage of the messages when diagnosing problems and is planned to be implemented. Detailed metrics of the Django program is also planned to be implemented for performance monitoring. This will ensure Data Central is able to adapt to the growing data volumes of data releases.

5. Summary

An immature ingestion system component for Data Central's ingestion system required a run time of 8 days to ingest data products for a data release. This was significantly larger than the required run time for other ingestion system components. Upon profiling the code with `py-spy` and monitoring the PostgreSQL database to identify and address the bottleneck, the runtime for ingesting blob data products was improved by a factor of 8. Additional profiling and monitoring solutions are planned to be implemented within Data Central's ingestion system to ensure Data Central adapts to the expanding size of data releases.

References

Gregg, B. 2016, Commun. ACM, 59, 48–57. URL `https://doi.org/10.1145/2909476`

Mannering, E., & O'Toole, S. 2019, in Linking Galaxies from the Epoch of Initial Star Formation to Today, 11

Astronomical Data Analysis Software and Systems XXXV
*ASP Conference Series, Vol. **VOL** Compiled: Tuesday 30th June, 2026@ 01:30*
K. Polsterer, D. Bohlender, S. Gwyn, A. Hincks and P. Teuben, eds.
*© **YEAR** Astronomical Society of the Pacific*

ZZZDEBUG!084.tex
Tocknell, J.|textbf
Miszalski, B.
software!development!DevOp
observatories!operations
observatories!ground-based
African Large Telescope
(SALT)
software!support
software!operations
software!project
tracking!JIRA

Adapting Telescope Operations to Science Platforms: How your organisation could benefit from a DevOps Roster

James Tocknell[1,2] and Brent Miszalski[1,2]

[1]*Australian Astronomical Optics, Macquarie University, North Ryde, NSW, Australia;* james.tocknell@mq.edu.au

[2]*Astronomy and Space Technology Research Centre, Macquarie University, North Ryde, NSW, Australia;* james.tocknell@mq.edu.au

Abstract. Whilst the Research Data and Software (RDS) team at the AAO has more than doubled in size, the project-based nature of our funding is such that there is very limited scope to hire additional infrastructure support personnel. Taking inspiration from the Southern African Large Telescope (SALT) Astronomy Operations user support model, we have introduced the concept of a "DevOps Roster" to ameliorate the load of growing operational tasks within our group. The roster aims to spread the support load over the larger team, while simultaneously upskilling the team with an emphasis on documenting gaps in our processes. The introduction of the roster has also generated interesting discussions about the long term funding and maintenance of operational tasks. In this talk proceedings, here we cover what has worked, possible pitfalls of such a model, and how other groups can benefit from its application.

1. Introduction and Context

The RDS (Research Data and Software) team within the AAO[1] at Macquarie University currently comprises of 20-odd staff working across the domains of software, data, project management and infrastructure support. RDS is almost entirely project funded, with individuals working in a matrix model across multiple projects. However, due to being embedded within a university and its human resource system, having sufficient dedicated support and infrastructure engineer roles (not tied to a specific project and instead manage common infrastructure) is not currently possible. An interim workaround was to use existing team members who already had the right skillset e.g. system or network administration for infrastructure, or sufficient knowledge of the numerous systems to triage user support tickets. This was not a long term solution to the missing dedicated roles, as instead the support burden disproportionately falls on specific individuals, reducing their effectiveness on projects and increasing the risk of burnout. Additionally, the bus factor of systems was low, commonly one.

[1]Australian Astronomical Optics, previously known as the Australian Astronomical Observatory and the Anglo-Australian Observatory.

2. Creation of the "DevOps Roster"

To provide a more sustainable solution, RDS introduced the concept of a "DevOps Roster", heavily derived from the SALT (Southern African Large Telescope) Astronomy Operations user support model (Väisänen et al. 2016) and adapted for the different context. The model at SALT has a rotating roster where staff focus on answering user helpdesk questions and preform daily tasks (e.g. data inspection, run pipelines, document issues).

The model RDS adopted has staff act as initial triage for user support, but also (as a lower priority role) complete tasks/tickets that have been assigned to the roster. The roster has 2 staff assigned per week to ensure there is a backup person if one becomes unavailable. At the end of the week the assignees complete the log of what work was done, any issues seen, and any ongoing issues for next week's roster. Significantly, it is not an on-call roster. Whilst the RDS team is spread out across Australia and so may incidentally provide support outside the 9am to 5pm common working hours, there is no expectation that those on the roster respond outside their normal working hours. This model was implemented at the beginning of 2025 and the objectives of the roster were to spread the support load widely to reduce burnout, document all existing systems, and to recognise and track system-wide support and maintenance. A suitable analogy may the service time of academic staff.

Before introducing the roster, most members of the team were not familiar with all the services RDS provides, especially all the components of the Data Central Science Platform. Now nearly all services have multiple people who can fix issues. The DevOps roster provides an additional path for mentoring junior members and introducing new members to the team. The roster has also reduced risks around the loss of institutional knowledge and allowed staff to take leave and not worry about needing to be contactable.

3. Tasks of the roster

Triage[2] of user helpdesk tickets is the highest priority, but there are many tasks and tickets which are assigned to be completed as part of the roster. These tasks vary in both time and the skillset required to complete or progress the task. While all those on the roster are engineers, different individuals will be more effective at completing certain tasks than others. For example, while some tasks require system or network administration knowledge, other tasks require detailed knowledge of specific systems, or a background in astronomy to complete.[3] The most common tasks usually fall within one of the following categories:

- Document how to handle common issues, such as users forgetting usernames and/or passwords, or users needing to migrate institutional emails.

[2]Those on the roster are not expected to complete all tickets on their own, it is expected and encouraged that they pull in additional assistance as required.

[3]The background of RDS members vary, with about 50% having a background in astronomy and the other 50% having an engineering background. The skillset of the team is also diverse, from general astronomy and data reduction pipelines, to hardware and instrument control to front-end web development, data science, systems engineering and network and security engineering.

- Minor changes to systems or develop a small web application to automate a task.
- Maintenance of systems (e.g. software updates).
- Major changes or platform-wide automation.

The primary tools of the roster are Slack (for coordination and discussion), with Jira tracking tickets (including the helpdesk) and Confluence providing a place for documentation.

Whilst some of these tasks are not what "DevOps" is sometimes interpreted as, e.g. a rebranding of system administration, the roster more closely matches the cultural aspects of breaking down silos and knowledge sharing associated with the original intent of DevOps and is more clearly emphasised under SRE (Site Reliability Engineering (Beyer et al. 2016)).

4. Putting the roster to the test: moving campuses

From July to November 2025 the AAO moved from our old building in North Ryde[4] to the main Macquarie University campus[5], a distance of approximately only 3–5 km. However, that small distance does not indicate the significant change in infrastructure that occurred. Previously much of the network and IT infrastructure was managed in-house by the AAO i.e. the RDS team, with an direct outside connection. On the main campus the majority of the network and IT infrastructure is run by Macquarie University Central IT. Across two major shutdown events, specific groups of servers and associated infrastructure were physically transported from one site to the other. During these shutdown events, those scheduled on the roster monitored the various user support channels. Prior to the move the roster had been in place for several months. The whole team were therefore familiar with the numerous services. This helped enormously when it came to the move, with respect to which services would be down due to each specific shutdown. Those on the roster were then able to ensure that only the services that were expected to be down were down. The roster also provided a clear process through which the rest of the AAO could be assisted with migrating their IT systems to campus.

5. Future of the roster and applying it to other groups

The roster has been very successful for the RDS team:

- Response and resolutions times of tickets have decreased.
- Internal tools have been developed to resolve common tasks, such as handling 2FA code resets.
- Continuous incremental progress has been made on large tasks (e.g. only one system still needs manual TLS certificate changes).

[4]The AAO resided in this building complex with the National Measurement Institute, as the AAO was previously part of the same Australian government department.

[5]Officially titled Macquarie University Wallumattagal Campus

There are no major changes planned for the roster currently, though performing a retrospective after a year of the roster to see what is working and what is not should clarify any possible changes and improvements.

We hope that other groups find something valuable to use in their context from our processes, and our suggestions for how to start such as roster are as following:

- Have buy-in from leadership
- Identify team members who should be on the roster:
  - Should be as wide as possible
  - Should include members of various skillsets and backgrounds
- Document how the roster works and have a few people to create and update the roster
- Have a place for knowledge sharing

On a final note, after the talk, the speaker (James) had conversations with staff from different observatories, who were in a similar situation to RDS before the introduction of the roster. This suggests that conversations within observatories need to be had (especially by leadership) around how to provide support, and how to communicate this effectively to funders and stakeholders.

Acknowledgments. We would like to acknowledge the rest of the RDS team for their feedback on the talk, and for being such excellent colleagues. We would also like to acknowledge those individuals with whom they had further discussions on the topic of support following the talk.

References

Beyer, B., Jones, C., Petoff, J., & Murphy, N. R. 2016, Site Reliability Engineering: How Google Runs Production Systems. URL `http://landing.google.com/sre/book.html`

Väisänen, P., Coetzee, C., Schroeder, A., Crawford, S. M., Buckley, D. A. H., Brink, J., Depagne, É., Kniazev, A., Koen, T., Kotze, P., Marang, F., Miszalski, B., Romero Colmenero, E., Strydom, O., Van Wyk, V., & Williams, T. B. 2016, in Observatory Operations: Strategies, Processes, and Systems VI, edited by A. B. Peck, R. L. Seaman, & C. R. Benn, vol. 9910 of Society of Photo-Optical Instrumentation Engineers (SPIE) Conference Series, 99100T

Astronomical Data Analysis Software and Systems XXXV
*ASP Conference Series, Vol. **VOL** Compiled: Tuesday 30th June, 2026@ 01:30*
K. Polsterer, D. Bohlender, S. Gwyn, A. Hincks and P. Teuben, eds.
*© **YEAR** Astronomical Society of the Pacific*

ZZZDEBUG!085.tex
Tocknell, J.|textbf
databases!Google BigQuery
databases!Microsoft SQL Server
databases!SAP iQ
databases!MySQL
databases!Postgres
databases!Postgres!Citus
databases!Postgres!Greenpl
methods!indexing!Hierarchi Triangular Mesh (HTM)
methods!indexing!PgSphere
methods!indexing!Q3C
software!frameworks!Spark
computing!cloud!databases
Q3C! ascl:1905.008

The state of clustered database solutions for astronomical data

James Tocknell[1,2]

[1]*Australian Astronomical Optics, Macquarie University, North Ryde, NSW, Australia;* james.tocknell@mq.edu.au

[2]*Astronomy and Space Technology Research Centre, Macquarie University, North Ryde, NSW, Australia;* james.tocknell@mq.edu.au

Abstract. The largest astronomical catalogues now exceed the capacity of a single machine, hence multiple groups have been experimenting with clustered database solutions to be able to scale past a single machine. This BoF provided a forum for sharing what solutions work and do not work, issues around licensing and possible solutions, and how to take existing astronomical database extensions and use them in clustered databases. This BoF aimed to start a wider collaboration to ensure that working solutions are maintained and can be effectively used by the whole community.

1. Introduction

This BoF started with the definition of an "Astronomical Clustered Database", namely any system satisfying the following three properties:

1. Astronomical—has support for spherical coordinates and indices (or extensions available). This is vital for doing positional queries, conesearches and cross-matches.
2. Clustered—has multi-node support and can shard tables (to handle multi-TB or larger tables). This mean read-replicas do not count as clustered.
3. Off-the-shelf database—can be SQL/NoSQL/NewSQL (though SQL is desirable) and not a custom/bespoke system (such as HATS (Caplar et al. 2025) or HiPS (Fernique et al. 2017)).

The purpose of this definition was to focus the discussion on how to provide a way to scale existing systems and solutions (e.g. TAP (Dowler et al. 2019)) and avoid spending time discussing systems (e.g. time-series databases) that would not meet the needs of users.

The motivation for holding this discussion at the end of 2025 was due to what seemed to have been the natural choice of the open-source Greenplum[1], however Broadcom bought VMWare (who had acquired Greenplum previously) and took it closed source.

[1]Greenplum is built on PostgreSQL and hence spherical coordinates and indices can be provided by Q3C (Koposov & Bartunov 2006) or pgSphere (Chilingarian et al. 2004)

1.1. Why Cluster?

The criteria for whether clustering should be used was most eloquently put by Shiny Brar as if the local storage more than twice as slow as the network, which given the size of both individual tables (up to TBs) and overall database size (PBs) and the current costs of storage and networking options, most astronomical data centres and science platforms will need some level of database clustering. Assuming the use hard discs[2] and 10 or higher gigabit interconnect between nodes[3], the local storage would be at least a factor of 3 slower than the network hence the clustering push.

2. Solutions Discussed

The solutions discussed can be placed into the following categories:

- PostgreSQL extensions and forks (which dominated the discussion);
- On-premise and cloud commercial SQL databases;
- Job/queue systems;
- Custom solutions.

The primary custom discussed was QServ (Mueller et al. 2025) coming out of LSST, which combines MySQL and Xrootd to be able to shard tables across multiple machines. Being custom however means that supporting other groups was not something its developers felt they capacity to provide, and the limitations of QServ around updates have meant that LSST has been experimenting with other databases solutions for their prompt product database. These include Google's BigQuery, which was found to be fast for conesearches, but for both generic queries and crossmatching the query time was excessive.

The commercial databases in use for astronomical that were discussed were Microsoft SQL Server (which has support for geometric queries) and SAP iQ (which lacks any specific geometric or astronomical query support). Both of these are in use by ESO, and their choice was due to their existing usage of these systems, with astronomical query support being added on top of SAP iQ through the use of HTM (Kunszt et al. 2001) to create spherical indices.

For job/queue systems, the primary solution discussed was Apache Spark, with ESA experimenting with Spark absent the rest of the Hadoop ecosystem for large queries. Other groups have previously examined or used the Hadoop ecosystem, but its limitations, along with the collapse of the Hadoop ecosystem, have caused groups to migrate away from it (Tocknell 2025).

2.1. Cloud-based solutions

BigQuery is not the only cloud-based managed database solution that exist, however cloud-based solutions were identified as having three key limitations:

[2]SSD/NVMe drives currently cost an order of magnitude more than HDDs, whilst tape is far too slow for what is effectively hot storage.

[3]Most server class hardware comes with 10 gigabit by default.

1. Cost of cloud solutions, both in terms of monetary value but also the misalignment between the nature of grant funding and the operation expense of cloud solution. Additionally some groups rather than purchasing hardware directly apply for resources on national facilities, and hence are not able to purchase cloud-based solutions.

2. Data Sovereignty

3. The lack of existing astronomical support from cloud-based astronomical solutions.

It seems unlikely that any of these limitations will be addressed soon, and so while "why not cloud" will likely continue to be raised in discussions like these, it seems that other than LSST no other group is considering using cloud-based solutions[4].

That does not mean on-premise or self-managed cloud (i.e. Kubernetes) solutions are not being considered, and there was some discussion about what to replace Bitnami OCI images and Helm charts with. For PostgreSQL, CNPG in use at both LSST and Data Central was the recommended replacement.

2.2. PostgreSQL-based solutions

PostgreSQL-based solutions have tended to be the benchmark against which other solutions have been measured. PostgreSQL combined with Q3C (Koposov & Bartunov 2006) and/or pgSphere (Chilingarian et al. 2004) provides a system allowing table uploads (which allows for crossmatching against external sources), spherical coordinates and indices for faster queries, alongside a well-maintained, efficient and vendor-neutral core with a expansive ecosystem. However, PostgreSQL itself does not come with native sharding support, and so therefore the challenge (as was discussed extensively in the BoF) is what parts of the ecosystem to use. For sharding, it would appear there are two types of solutions: PostgreSQL extensions and PostgreSQL forks. The leading PostgreSQL extension solution is currently Citus, whilst the previous leading PostgreSQL fork solution was Greenplum, though the change to go closed source has spawned multiple forks, open source or otherwise, to try become the leading solution.

While both Citus and Greenplum use similar concepts, the differences between the solutions emphasise the tradeoffs between choosing a PostgreSQL fork and a PostgreSQL extension. Citus rapidly supports the latest PostgreSQL release[5], while Greenplum in its last open release is currently six major versions of PostgreSQL behind. On the other hand, the choice to fork PostgreSQL allowed Greenplum (and its forks) to provide more cohesive experience and easier to use tooling.

Current uses of Citus solutions include Data Central and the Japanese LSST IDAC[6], with CADC planning on adopting a Citus-based solution for their TAP system. Whilst

[4]The absence of many US-based groups, where cloud usage is the highest, may be the reason no other group in the room is or plans to use cloud-based solutions

[5]As of the submission of this proceedings, Citus supported PostgreSQL 17, with support for PostgreSQL 18 coming in the next imminent Citus release.

[6]Data Central runs a more traditional 4 node citus cluster with HDDs and 10G interconnect, whilst the Japanese LSST IDAC is running multiple instances of PostgreSQL on a single powerful node with NVMe, which could be quite revolutionary.

Greenplum had been used by multiple groups, no Greenplum-fork-based system was presented as a clear solution.

However, whilst in theory being built on top of PostgreSQL would allow use of the various extensions together, this is sadly not the case. Due to current limitations of how Q3C provides estimates to the PostgreSQL planner, on both Greenplum-based solutions and Citus, PostgreSQL is unable to effectively take full advantage of spherical indexing, resulting in slower queries. It was brought up that while Q3C has traditionally been at least an order of magnitude faster than pgSphere, improvements in pgSphere have resulted in the performance difference between pgSphere and Q3C shrinking to at worst a factor of 2, and pgSphere does not have the same limitations around query estimates.

3. Summary and Outcomes

From the discussion, it seems the current best open source solution is Citus with pgSphere, but that modifications to Q3C or convergence on a single Greenplum fork may change that consensus. Additionally, the has been the creation of a mailing list to enable private discussions between the database engineers and support staff at the different institutes to share findings about different solutions and products on an ongoing basis.

Acknowledgments. The author would like to acknowledge all the participants of the BoF for their excellent contributions to the discussion.

References

Caplar, N., Beebe, W., Branton, D., Campos, S., Connolly, A., DeLucchi, M., Jones, D., Juric, M., Kubica, J., Malanchev, K., Mandelbaum, R., & McGuire, S. 2025, arXiv e-prints, arXiv:2501.02103. eprint: 2501.02103

Chilingarian, I., Bartunov, O., Richter, J., & Sigaev, T. 2004, in Astronomical Data Analysis Software and Systems (ADASS) XIII, edited by F. Ochsenbein, M. G. Allen, & D. Egret, vol. 314 of Astronomical Society of the Pacific Conference Series, 225

Dowler, P., Rixon, G., Tody, D., & Demleitner, M. 2019, Table Access Protocol Version 1.1, IVOA Recommendation 27 September 2019

Fernique, P., Allen, M., Boch, T., Donaldson, T., Durand, D., Ebisawa, K., Michel, L., Salgado, J., & Stoehr, F. 2017, HiPS - Hierarchical Progressive Survey Version 1.0, IVOA Recommendation 19 May 2017. eprint: 1708.09704

Koposov, S., & Bartunov, O. 2006, in Astronomical Data Analysis Software and Systems XV, edited by C. Gabriel, C. Arviset, D. Ponz, & S. Enrique, vol. 351 of Astronomical Society of the Pacific Conference Series, 735

Kunszt, P. Z., Szalay, A. S., & Thakar, A. R. 2001, in Mining the Sky, edited by A. J. Banday, S. Zaroubi, & M. Bartelmann, 631

Mueller, F., Gaponenko, I., Gates, J., Hanushevsky, A., Jammes, F., Lim, K.-T., Salnikov, A., & Slater, C. T. 2025, in Astronomical Data Analysis Software and Systems XXXII, edited by S. Gaudet, D. Bohlender, S. Gwyn, A. Hincks, & P. Teuben, vol. 538 of Astronomical Society of the Pacific Conference Series, 114

Tocknell, J. 2025, in Astronomical Society of the Pacific Conference Series, edited by A. Jacques, R. Seaman, N. Gandilo, & T. Linder, vol. 541 of Astronomical Society of the Pacific Conference Series, 458

Astronomical Data Analysis Software and Systems XXXV
*ASP Conference Series, Vol. **VOL** Compiled: Tuesday 30th June, 2026@ 01:30*
K. Polsterer, D. Bohlender, S. Gwyn, A. Hincks and P. Teuben, eds.
*© **YEAR** Astronomical Society of the Pacific*

ZZZDEBUG!086.tex
Wang, Y.|textbf
Cui, C.
Tao, Y.
Mi, L.
He, B.
FOOBAR!conference!ADASS 2019
FOOBAR!organisations!ASP
FOOBAR! ascl:1101.010

The NADC Dataset Metadata Specification: Enabling FAIR Astronomical Data Discovery

Youfen Wang,[1] Chenzhou Cui,[1] Yihan Tao[1], Linying Mi,[1] and Boliang He[1]

[1]*National Astronomical Observatories, CAS, Beijing, Chaoyang, China;*
yfwang@bao.ac.cn

Abstract. This document establishes the NADC (National Astronomical Data Center) Metadata Specification for Astronomical Science Datasets, defining description methods and extension rules for 85 metadata elements categorized into three functional subsets: Identity (9 elements), Curation (49 elements), and Content (27 elements). The specification adopts a three-tier hierarchy—Core Metadata (28 elements), General Metadata, and Specialized Metadata—aligned with GB/T 30523-2023 and international standards including IVOA, DOI, and Dublin Core. It standardizes persistent identifiers (IVOA, DOI, CSTR), provenance tracking, and technical specifications for diverse data types including telescope observations, simulations, and mirror datasets. Applicable to the complete data lifecycle (submission, registration, cataloguing, archiving, publication, and sharing), this standard ensures interoperability, discoverability, and proper attribution of astronomical scientific resources within NADC-managed repositories.

1. Scope

This document specifies the description methods, composition, elements, entities, and extension types and rules for metadata of astronomical science datasets at the National Astronomical Data Center (NADC).

This document is applicable to the submission, registration, cataloguing, archiving, publication, sharing, exchange, and query of astronomical science datasets managed by NADC.

2. Normative References

The following referenced documents are indispensable for the application of this document. For dated references, only the edition cited applies. For undated references, the latest edition of the referenced document (including any amendments) applies.

- GB/T 30522-2014, *Basic principles and methods for metadata standardization of science and technology platforms*
- GB/T 30523-2023, *Core metadata for science and technology resources*
- GB/T 32843-2016, *Science and technology resource identifier*
- GB/T 13745-2009, *Classification and code of disciplines*

- IVOA Identifiers Version 2.0
(`https://www.ivoa.net/documents/\penalty0IVOAIdentifiers/`)

3. Terms and Definitions

3.1 metadata data that define and describe other data
[Source: GB/T 18391.1-2009, 3.2.16]

3.2 astronomical scientific data collective designation for observational data obtained through observation, simulation, and other means during astronomical research activities, as well as various derived data produced through systematic processing and treatment

3.3 dataset a collection of data with a specific theme, identifiable and capable of being processed by computers
[Source: GB/T 31075-2014, A.2.16]

3.4 metadata element basic unit of metadata
[Source: GB/T 30523-2023, 3.4]

3.5 metadata entity group of metadata elements that describe the same characteristic of data
[Source: GB/T 30523-2023, 3.5]

3.6 core metadata minimal set of metadata describing the most basic information of data
[Source: GB/T 30523-2023, 3.3]

3.7 identity metadata metadata describing dataset names, identifiers, and other identification information

3.8 curation metadata metadata describing data maintenance and publication information (e.g., version, release date)

3.9 content metadata metadata describing the content information contained in the data

Abbreviations

CSTR	China Science and Technology Resource Identifier
DOI	Digital Object Identifier
IVOA	International Virtual Observatory Alliance

4. Composition and Description of Astronomical Science Dataset Metadata

4.1. Content Composition

According to content and function, astronomical science dataset metadata is divided into three metadata subsets: Identity Metadata, Curation Metadata, and Content Metadata. Each subset comprises metadata elements and entities that collectively describe specific classes of dataset attributes.

Identity Metadata includes metadata entities or elements such as Chinese Title, English Title, Identifier Information, and Short Name.

Curation Metadata includes metadata entities or elements such as Dataset Source Type, Dataset Status, Version, Personnel Information (Creator, Contributor, Publisher, Service Provider), Time Information, Access Information, Copyright and Citation Information, Statistics, and Relationship Information.

Content Metadata includes metadata entities or elements such as Chinese Description, English Description, Keywords, Subject Classification, Processing Level, Data Format, Observation Band, Facility, Coverage, and Resolution.

4.2. Hierarchical Architecture

Based on applicability scope, astronomical science dataset metadata is classified into three hierarchical levels: Core Metadata, General Metadata, and Specialized Metadata.

Core Metadata represents the minimal set of basic metadata required for dataset publication and sharing, primarily referencing the requirements of GB/T 30523-2023 *Core Metadata for Science and Technology Resources*, as well as general requirements from the China Science and Technology Resource Sharing Network, China Science and Technology Resource Identifier (CSTR), Digital Object Identifier (DOI), and International Virtual Observatory Alliance (IVOA).

General Metadata represents an extended set built upon Core Metadata, used to more comprehensively describe astronomical-specific attributes of datasets.

Specialized Metadata represents metadata describing proprietary attributes for specific types of astronomical datasets within the domain.

4.3. Description Method

This document defines and describes astronomical science dataset metadata using the abstract representation method specified in GB/T 30522, including: definition, English name, data type, value domain, short name, annotation, sub-elements, and Extended Backus-Naur Form (EBNF).

5. Metadata Elements

The NADC Astronomical Science Dataset Metadata Specification defines a total of 85 metadata elements across three subsets. Detailed Breakdown by Category:

- Identity Metadata (9 elements):

  Chinese Title, English Title, IVOA Identifier, IVOA Registration Date, DOI, DOI Registration Date, CSTR Identifier, CSTR Registration Date, and Short Name.

- Curation Metadata (49 elements):

  Dataset administration (3): Dataset Source Type, Dataset Status, Dataset Version

  Personnel information (24): Four roles (Creator,

  Contributor, Publisher, Service Provider) each comprising elements (Name, Affiliation, Email, Phone, Address, Postcode)

Time information (4): Creation Date, Publication Date, Metadata Created, Metadata Updated

Access information (5): Share Method, Share Scope, Application Procedure, Storage Path, URL

Copyright and citation (4): Usage Guide, Intellectual Property Category, License, Acknowledgement

Statistics (3): Data Volume, File Number, Record Number

Relationship information (6): Parent Dataset, New Version, Derived Dataset, Project ID, Paper ID, Software ID

- Content Metadata (27 elements):

  Descriptive elements (3): Chinese Description, English Description, Keywords

  Subject classification (2): Subject, Product Type

  Technical specifications (4): Processing Level, Data Format, Observation Band, Facility

  Coverage (14): Coverage (composite entity), Spatial Coverage, Spectral Coverage (with 4 sub-elements: Band Name, Central/Minimum/Maximum Wavelength), Temporal Coverage (with 2 sub-elements: Start/Stop Time), Coverage Depth, Object Density, Object Count, Sky Fraction

  Resolution (4): Resolution (composite entity), Spatial Resolution, Spectral Resolution, Temporal Resolution.

Among the 85 metadata elements:

- **28 elements** constitute the **Core Metadata** (minimum set required for publication).
- **78 elements** constitute the **General Metadata** (which includes the 28 Core Metadata elements plus 50 additional general elements).
- **7 elements** constitute the **Specialized Metadata** (domain-specific attributes).

The complete information of the metadata elements, including definition, English name, data type, value domain, short name, annotation, sub-elements, and Extended Backus-Naur Form (EBNF), is available in the supplementary tables at: `https://nadc.china-vo.org/res/r101767/`.

Acknowledgments. This work is supported by the National Key R& D Program of China No. 2022YFF0711500.

Astronomical Data Analysis Software and Systems XXXV
*ASP Conference Series, Vol. **VOL** Compiled: Tuesday 30th June, 2026@ 01:30*
K. Polsterer, D. Bohlender, S. Gwyn, A. Hincks and P. Teuben, eds.
*© **YEAR** Astronomical Society of the Pacific*

ZZZDEBUG!088.tex
Arviset, C.|textbf
Merín, B.
Bhatawdekar, R.
Baines, D.
Bentley, M.
de Bruijne, J.
de Marchi, G.
Masson, A.
Navarro, V.
Reerink, J.
Perez, H.
Barreiro, M. F.
Cook, J.
Gil, P. P.
Andres, A.
Leon, I.
Docasal, R.
Norman, H.
Matsson, P.
FOOBAR!conference!ADASS 2019
FOOBAR!organisations!ASP
FOOBAR! ascl:1101.010

A 30-year journey through science data management

Christophe Arviset[1], Bruno Merín[1], Rachana Bhatawdekar[1], Deborah Baines[1], Mark Bentley[1], Jos de Bruijne[2], Guido de Marchi[2], Arnaud Masson[1], Vicente Navarro[1], Jan Reerink[1], Hector Perez[3], Monica Fernandez Barreiro[3], Jonathan Cook[3], Pablo Perez Gil[3], Anastasia Andres[3], Ignacio Leon[4], Ruben Docasal[4], Henrik Norman[5], Philip Matsson[5]

[1]*European Space Agency, ESAC, Spain;* Christophe.Arviset@esa.int

[2]*European Space Agency, ESTEC, The Netherlands*

[3]*Starion for the European Space Agency, ESAC, Spain*

[4]*Aurora for the European Space Agency, ESAC, Spain*

[5]*WinterWay for the European Space Agency, ESAC, Sweden*

.

Abstract. Do you remember which computer you were using in 1995? What were the favourite operating systems and programming languages in the ADASS community back then? The World Wide Web was just born, and it has changed everybody's life to a point nobody ever imagined. When ESA decided to build the ISO Data Archive in 1995, no one expected this would be the beginning of a journey towards establishing the ESAC Science Data Centre. ESA's Digital Library of the Universe is now covering more than 30 astronomy, planetary and heliophysics missions' science archives and their associated data discovery portal (ESASky) and its close integration into the data exploitation platform (ESA Datalabs). Information Technology is changing rapidly, and data archives also went through major changes over the last three decades, as well as their associated science data management hardware and software systems, including storage, databases, graphical user interfaces, data distribution, interoperability and now data exploitation through science platforms.

Hence, let's go back in time and go through a 30-year journey of science data management, highlighting the Lessons Learned along the way!

1. From 1995 to 2025

Looking at the world today where technology is evolving so quickly, it is worth looking 30 years back at what the state-of-the art devices were. Phones were landlines, used only to talk and were definitely not yet "smart". Apple revolutionized the human-computer interactions with a mouse and graphical operating system. We used to listen to music through a Walkman and NCSA Mosaic navigator enabled people to easily access static information in remote places. Who could have bet then that the Internet would become part of any moment of our daily life?

The first ADASS conference took place in 1991 in Tucson, USA where presentations were provided through physical handwritten slides (PowerPoint and KeyNote did

not exist then!), and we're now celebrating the 35th ADASS anniversary, demonstrating the sustainable interest and strength of this community over the last three decades, year after year, even despite some tragic worldwide events (9/11 tragedy in 2001 before ADASS in Victoria and the COVID pandemic in 2020 with the first full remote conference in "Granada") and some more joyful ones (Solar Eclipse in 2005 in San Lorenzo de El Escorial). The Virtual Observatory appeared for the first time as a concept in 2001 and has evolved through some bumpy roads to show its full potential at ADASS 2015 in Sydney, where new astronomical data centres reported VO implementation without having been involved in it before.

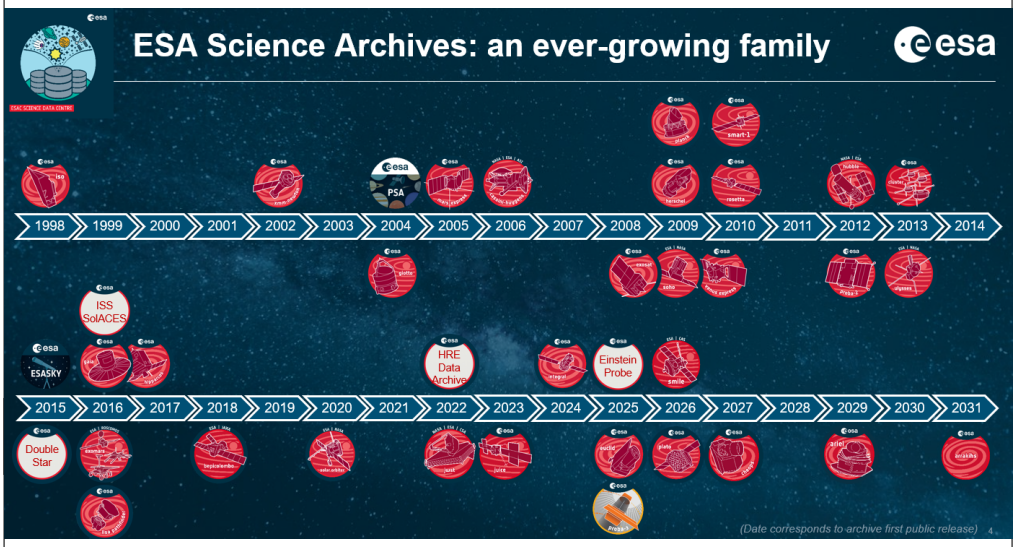


Figure 1. ESA Archives ever growing family.

ESA strategy for data management for its Science missions has also very much evolved over time. In 1998, around the end of its in-orbit operations, the first public archive for ISO (Infrared Space Observatory) was released on the web, through a modern Java applet GUI, offering easy and fast access to all ISO data products. It took some time for other missions to adopt the same path, with XMM-Newton science archive released in 2002 and the Planetary Science Archive, hosting data from various ESA planetary missions. Building on these successes, ESA decided to host all its Science missions data archives within the ESAC Science Data Centre[1] (ESDC), located in Villanueva de la Canada, close to Madrid, Spain. Understanding that data needs to be accessible to the scientific community as soon as available, Planck and Herschel were the first two archives to be ready at launch in 2009. Archives located in other places such as Cluster Active Archive and the Ulysses operational archive from ESTEC, Netherlands and the European HST Archive from ST-ECF, Garching, Germany were transferred to ESAC to ensure operations and maintenance within the same team, using similar technologies. Special efforts (almost "archeological" exploration) took place to gather data from very old missions (i.e. Giotto, Exosat, Ulysses).

[1] `https://archives.esac.esa.int`

Taking profit of having several astronomical data archives at ESAC, and building on existing data interoperability VO standards, ESASky[2] was released in 2015. Gaia was the first mission to trigger significant technology changes for its data management, due to the volume, complexity and variety of its data. Other directorates at ESA leveraged on ESDC experience for the HREDA[3] (Human and Robotic Exploration Data Archive) released in 2022, hosting all datasets from the experiments made on-board of the ISS (International Space Station), and through any other platforms. The upcoming big challenge is coming from Euclid, which data volume represents more than one order of magnitude greater than the total of all the other archives hosted at ESAC.

2. ESAC Science Data Centre in 2025

Overall today in 2026, ESDC is hosting more than 30 missions science archives covering all ESA Space Science fields (astronomy, planetary, heliophysics and low-earth orbit platforms) as shown in Figure 1. All these archives are being developed, operated, and maintained by the same team, comprised of archive scientists and software engineers, organized along the four science fields mentioned above.

ESA Science Archives Long Term Strategy is now based around three main pillars. First, enabling maximum scientific exploitation of Science datasets. ESA science missions' success is often being measured by the number of scientific papers written with mission data. Papers using archival data represent between 50% and 75% of them (De Marchi & Parmar 2025), so it is extremely important that science archives support this purpose. The Archive Scientist for the mission is responsible for gathering science cases and user requirements for their mission, and for working closely with the archive development team for the archive to ensure the archives efficiently serves the scientific community. Second pillar is to ensure efficient long-term preservation of data, software and knowledge. While missions can last over several decades, they always come to an end one day. But the science continues well after the mission lifetime, and it is ESA responsibility to ensure that the science archives from legacy missions continue to be accessible and to produce new scientific papers. Last but not least, the third pillar calls for the development of cost-effective archives via integration in, and across, projects. In some cases, the archive is part of the data processing chain of the mission, so needs to be smoothy integrated into the mission Science Ground Segment's overall architecture. In some other cases, the mission data is within a multi-mission data infrastructure, benefiting from existing functionalities and where specific missions' requirements also need to be factored in.

3. ESDC successes and Lessons Learned over the years

3.1. Science Archives are more than just FTP or JBOD

In many people's mind, Archives consist only of dumping all data from the missions onto some FTP server for the community, a JBOD in other words (Just a Bunch of Disks). But archives are much more than this, and are full data management services

[2] `https://sky.esa.int`

[3] `https://hreda.esac.esa.int/`

as indicated in the Open Archiving Information System (OAIS) functional model (Figure 2, reproduced with permission from the Consultative committee for Space Data Systems (CCSDS[4])).

From the back-end perspective, a lot of attention is paid to data curation, ensuring the best quality of the data being ingested into the archive, then extraction of metadata into the database to enable queries as per the science cases defined by the Archive Scientist. From the front-end perspective, the science archives offer more than just a list of products through FTP, but powerful data discovery and visualization services. These services allow scientists to search for particular regions of the sky or solar system objects, combining this with a wide variety of other search parameters based on science objectives. Visualization services also play an important role to help the scientists deciding what to retrieve from the archives.

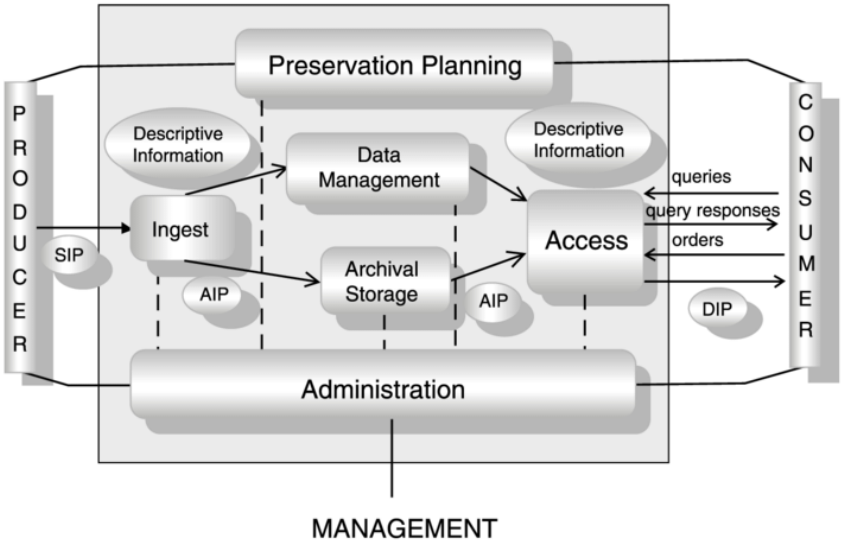


Figure 2. Open Archiving Information System functional model.

The administration part also represents a fair amount of work, with database administration, data and users access management, usage logging and reporting. And as archives must perdure over decades while technology evolves much more rapidly, preservation activities are required for technology evolution for all the archive subsystems. Another important aspect is the interoperability with other external archives services, delegating parts which are better done by other entities. The most scientifically useful example is the linking of data with their scientific publications. Allowing access to the scientific papers related to the data and vice-versa greatly enhances the usage of the data. While this linking was mostly done manually, there are real expectations that AI and NLP will help significantly in this process.

Overall, a science archive system can represent tens of thousands of lines of software code.

[4] https://ccsds.org/

3.2. Good architecture is the key and it evolves with time

From their conception in the mid-90s, the ESAC Science Archives architecture has been based on a 3-tier model (Data layer, Server layer and Client layer, see Figure 3). The modular approach has been one of the best decisions, as it has allowed the archives subsystems to evolve, without the need to redevelop them all at once.

From CDROM jukeboxes in 1998 to having all data available on hard disks for fast access, from commercial relational databases to open-source ones with community add-ins (pgsphere, postgis), from Java 1 to Java 18 for the server part, from Java Applet to Google Web Toolkit and now Angular for the GUIs, from interoperability with a VO translation layer toward a VO built-in approach, many evolutions have taken place smoothly over the last three decades (Bhatawdekar et al. 2026). This has also allowed the archives to easily connect the archives to new key tools and services, such as ESASky, Astropy or ESA Datalabs.

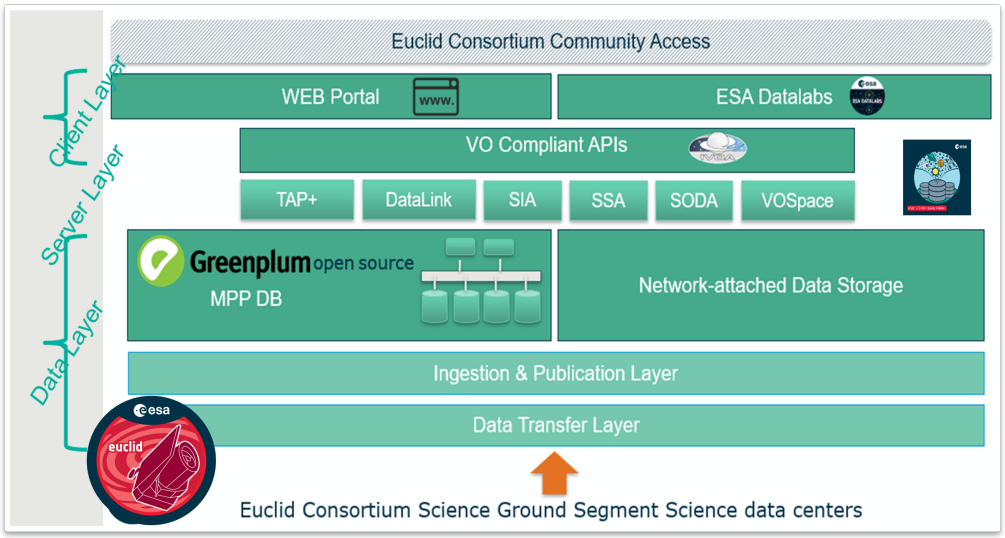


Figure 3. Euclid Archive 3-tier Architecture with VO built-in (Nieto et al. 2025).

3.3. ESASky: a window to the universe

While the number of astronomical archives was growing at ESAC, we established the first ESA Science Archives Long Term Strategy in 2013. This was the result of intense brainstorming and discussions among scientists and software engineers within our department, defining new science cases. One of the main outcomes was the need to build a multi-mission archive. As the VO data interoperability protocols were getting to the right level of maturity, with increased implementation across ESA science archives and other data centres worldwide, we decided to leverage these and existing tools (in particular Aladin Lite from the CDS, Centre des Données de Strasbourg) to build a web-based, science-driven, multi-mission astronomical data discovery portal with simple, intuitive, and unified access to data initially from ESA and later on from a wide range of space- and ground-based observatories worldwide. And ESASky was born and released in 2015 (Giordano et al. 2018).

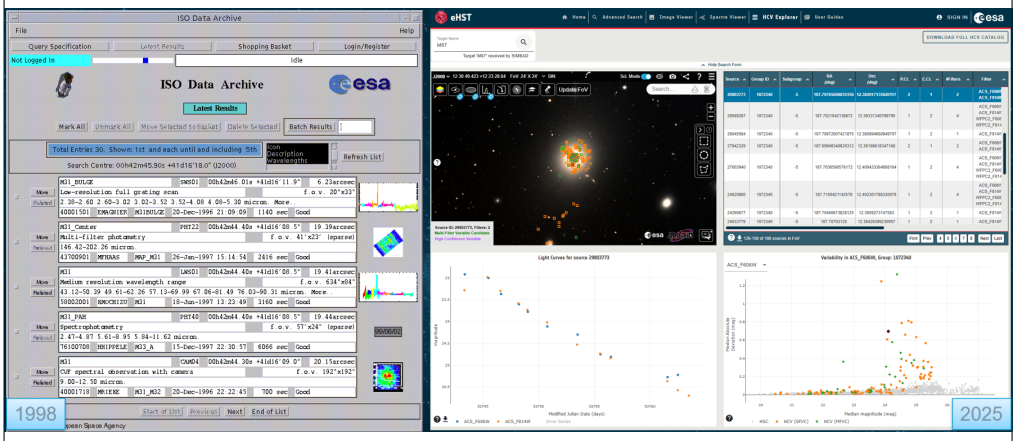


Figure 4. Evolution of Archives GUI evolution over time. *Left:* ISO Archive GUI in 1998. *Right:* European Hubble Archive in 2025.

Today, ESASky provides access to science-ready images, spectra, and catalogues from a broad range of ESA astronomy missions, alongside various space observatories data from NASA, JAXA and ground-based observatory from ESO, LAMOST, and SDSS datasets, enabling comprehensive multi-wavelength analysis. The tool is entirely web-based and mission-agnostic, requiring no prior knowledge of individual mission characteristics, thereby supporting both expert astronomers and non-specialists. ESASky employs advanced visualisation technologies such as HiPS all-sky mosaics and mission coverage maps (MOC) to efficiently display massive datasets. Users can interactively explore the sky, overlay footprints from multiple instruments, and compare observations across wavelengths in a unified environment. The portal supports multi-target searches, provides summarised observation metadata, and facilitates rapid retrieval of relevant data products. Additional capabilities include serendipitous searches of Solar System object observations and tools for observing, planning, including support for JWST preparation.

4. The Virtual Observatory

The International Virtual Observatory Alliance (IVOA) was created in 2002 with the primary mission to enable global, integrated access to astronomical data collected by observatories worldwide. The Alliance consists today of 23 members who participate at different levels (Arviset et al. 2025). There is no specific budget for the IVOA and participation has fluctuated depending on the funding availability from each of its members. But today, the IVOA is a very stable organization, with established working groups and interest groups, a clear process to define recommended data interoperability standards, and meetings twice per year with over 100 participants. Over the years, the funding went from dedicated national VO projects towards less specific VO funding but work integrated within the astronomical data centres worldwide. While it was important in the early days to have people brainstorming about new standards, it became later on more effective to have the active participation of the data centres to ensure the VO standards were taken up and VO data services were deployed.

From the early days, there might have been some misconceptions of the VO, which can explain why sometimes the VO was considered as not successful in the community. Contrary to what some people might have expected, the Virtual Observatory is not a portal to access all astronomical data worldwide; neither is it the ultimate application in astronomy nor a major solution for data management. Instead, the VO is a data interoperability ecosystem between astronomical data providers, an alliance of members defining standards to build VO applications for multi-observatories data access, and overall a data management framework for building interoperable data services.

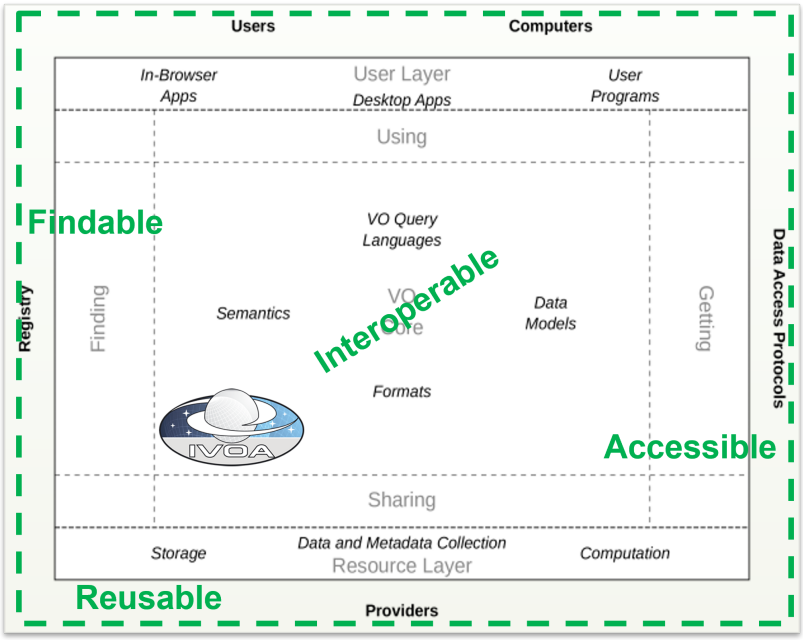


Figure 5. IVOA Architecture and FAIR.

The IVOA has tackled the data interoperability challenges well with an IVOA Architecture defined in 2010, which was FAIR before FAIR even existed (see Figure 5), and has been very stable since then. Most astronomical data today is accessible through VO protocols. Gaia is the perfect example, where data access leverages IVOA standards such as TAP, ADQL, UWS and Datalink and one can affirm that all astronomers using Gaia data are actually using the VO (and sometimes without knowing it). With the advent of big data and the associated science data platform, the next big challenge to address for the IVOA will be the smooth interoperability between these platforms.

5. Big upcoming challenges and opportunities

5.1. User login or not login: that is the question and a changing answer

In 2000, user login was required for all ESA science archive services. All archives had then a mix of proprietary and public data, mostly available through Graphical Web User Interfaces. Querying for data and metadata did not require login, but any data download

request required user authentication and authorization. Archive usage statistics were then easy as all data downloads were associated with a user account.

In 2010, some of the ESA science archives only had public data, and alongside the usual Web GUI, APIs offered convenient scriptable access to the data. User login then became optional and only required when accessing proprietary data, on-the-fly reprocessed data and large data bundle (>a few GB). Archive usage statistics were then based both on users (when logged in) and IP addresses (for anonymous access), with a gradual increase of the latter over the former.

In 2016 with the first Gaia public data release, while all data was public the user login was reinforced to gain access to more advanced user services such as bigger data queries or user workspaces within the archive, but login was not required for standard services. Archive usage statistics continued to cover logged in user accesses and anonymous ones, with a clear bigger portion of the latter.

In 2025, while the amount of public data greatly exceeds the proprietary data and anonymous access through GUI or API represents the majority of accesses, we see our archive usage statistics polluted by Bots and LLM crawlers, with accesses which can be 1000 times bigger than the standard usage. While totally distorting the access statistics represents the risk of unrealistic and unreliable statistics, we also start to fear potential degradation of our services towards our real scientific community, due to other "users". Providing more advanced data services through an exploitation platform is another reason why re-instantiating user login for any data download is seriously considered for all our ESA science archives, as we had 25 years ago.

5.2. Long Term Data Preservation Challenges

Science missions usually span over several decades through their study, development, in-orbit operations and post-operations phases, during which dedicated mission budget is available and a small portion is for the development and operations of the mission science archive. When the mission enters into its legacy phase, there is no more dedicated funding for this mission, but the archive still needs to be maintained and operated, as many scientists continue to access its data to advance science, through refereed papers, value added products (e.g. catalogues) or educational purposes. Some limited funding is available within ESA through the Heritage Space Programme to cover the archives for missions in legacy. As all archives are being maintained by the same ESAC Science Data Centre team, these tasks can be done in a cost efficient manner as the same software developer continues to operate the legacy archives, while developing the ones for the new missions. But the archive cannot be totally frozen, as technology evolves, and legacy archives need migration from old technology to new ones (new database versions, more modern back-end frameworks and front-end GUIs).

The number of archives in legacy will outnumber the ones of active missions, which is forcing us to reconsider the approach of having a single archive per mission. The Planetary Science Archive (PSA) already integrates all the planetary missions regardless of their phase and we're going to implement the same model for other science disciplines. A first public version of the HelioPhysics Archive (HPA, see Figure 6) has been released in April 2025, starting with an advanced Field of View tool for the Solar Orbiter data, followed by a first version of a totally re-designed Solar Orbiter archive (July 2025). The two most recent heliophysics missions, Proba-3 and SMILE will have their archives directly released in HPA during the course of 2026, together with two legacy archives (ISS-SolACES and Ulysses). The rest of the heliophysics missions will

gradually be moved to HPA. Similar approach has also started with astronomy missions with the building of the Astronomy Multi-missions Archive (AMA), although the wide variety of astronomy missions and associated archive services makes it more challenging.

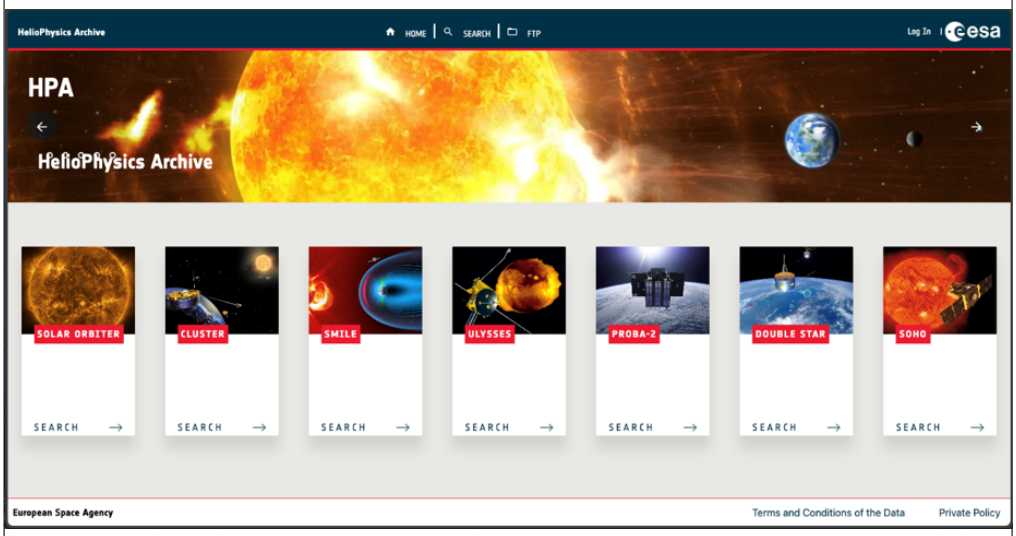


Figure 6. ESA HelioPhysics Archive.

5.3. ESA Datalabs and the new data access paradigm

The total volume of all ESA science archives up to 2024 was about 1PB, but Gaia and Euclid more than triple this amount by 2026 and Euclid (de Bruijne & Altieri 2025) will bring it to another order of magnitude by the beginning of the next decade. The big data era is now a reality, and not just because of the data volume, but also due to the variety and change velocity of data and metadata. For some of the new missions, Euclid in particular, the scientists will not be able to download the data to their premises to do their science. Hence, a paradigm shift is necessary: from a "bring the data to the user" approach towards "bring the user to the data", through the development of science data exploitation platforms. ESA has been developing a platform called ESA Datalabs (Navarro et al. 2024), closely integrated with the existing archives. Beyond delivering this paradigm shift for new big data missions, it will also leverage AI and ML techniques to enable new data discoveries. Furthermore, ESA Datalabs offers a Virtual Collaboration Environment for scientists, where people can share their data and their code into a common workspace, and eventually publish their analysis software on the public applications store of ESA Datalabs. Moreover, ESADatalabs supports preservation of analysis software from legacy missions. This code can be published into the platform, directly available to end users, hence facilitating its installation and ensuring its long-term preservation and availability.

6. Conclusions

The ESAC Science Data Centre has been developing and operating over the last 30 years more than 30 science archives from ESA Space Science missions. More than just data repositories, the science archives are full data management systems that enables the scientific return of the missions to be maximised. Their robust and modular architecture has ensured their maintainability and has facilitated their technology evolution over time. The VO has been key for data interoperability among datasets and astronomical data partners worldwide. The next strategic step towards ensuring long-term sustainability will be to move from a dedicated archive for each mission towards multi-mission archives per scientific discipline. The big data era has arrived and the science data platforms such as ESA Datalabs will be key to ensure that the ESA Science Archives continue to serve best the scientific community in the best possible way.

Acknowledgments. The authors would like to thank all the people who are or who have been part of the ESAC Science Data Centre over the years, as well as all supporting teams in particular the Programmatic IT section at ESAC, the missions' Science Operations Centres and Project Scientists, the Instrument Teams and missions' Consortia. They have been part of this data management journey over the last decades.

References

Arviset, C., O'Toole, S., IVOA Executive Committee, & Technical Coordination Group 2025, in Astronomical Society of the Pacific Conference Series, edited by A. Jacques, R. Seaman, N. Gandilo, & T. Linder, vol. 541 of Astronomical Society of the Pacific Conference Series, 68

Bhatawdekar, R., et al. 2026, in ADASS XXXV, edited by TBD (San Francisco: ASP), vol. TBD of ASP Conf. Ser., TBD

de Bruijne, J., & Altieri, B. 2025, in Astronomical Society of the Pacific Conference Series, edited by A. Jacques, R. Seaman, N. Gandilo, & T. Linder, vol. 541 of Astronomical Society of the Pacific Conference Series, 12

De Marchi, G., & Parmar, A. 2025, in ESA Science Programme Missions, edited by A. Parmar (Berlin: Springer Nature), vol. 18 of ISSI Scientific Report, 9–69

Giordano, F., Racero, E., Norman, H., Vallés, R., Merín, B., Baines, D., López-Caniego, M., Martí, B., de Teodoro, P., Salgado, J., Sarmiento, M., Gutiérrez-Sánchez, R., Prieto, R., Lorca, A., Alberola, S., Valtchanov, I., de Marchi, G., Álvarez, R., & Arviset, C. 2018, Astronomy and Computing, 24, 97

Navarro, V., del Rio, S., Diego, M. A., Lopez-Caniego, M., Marinic, F., Kruk, S., Reerink, J., & Arviset, C. 2024, ESA Datalabs: Digital Innovation in Space Science (Springer Nature Singapore), 1

Nieto, S., Altieri, B., López-Caniego, M., de Teodoro, P., Oliveira, J., Fernández, J., Alonso-Albi, T., Ríos, C., Pérez, P., de Bruijne, J., & et al. 2025, in Astronomical Society of the Pacific Conference Series, edited by A. Jacques, R. Seaman, N. Gandilo, & T. Linder, vol. 541 of Astronomical Society of the Pacific Conference Series, 28

Astronomical Data Analysis Software and Systems XXXV
*ASP Conference Series, Vol. **VOL** Compiled: Tuesday 30th June, 2026@ 01:30*
K. Polsterer, D. Bohlender, S. Gwyn, A. Hincks and P. Teuben, eds.
*© **YEAR** Astronomical Society of the Pacific*

ZZZDEBUG!090.tex
Parmiggiani, N.|textbf
Bulgarelli, A.
Panebianco, G.
Burns, E.
Neights, E.
Fioretti, V.
Martinez-Castellanos, I.
Castaldini, L.
Ciabattoni, A.
Di Piano, A.
Falco, R.
Gallego, S.
Mustafa, G.
Patel, P.
Rizzo, A.
Wulf, E. A.
Hartmann, D. H.
Kierans, C. A.
Tomsick, J. A.
Zoglauer, A.

Localization and Confidence Region Estimation of Short GRBs with the COSI BGO Shield Using a HEALPix-Based Deep Learning Approach

N. Parmiggiani,[1] A. Bulgarelli,[1] G. Panebianco,[1] E. Burns,[2] E. Neights,[3,4] V. Fioretti,[1] I. Martinez-Castellanos,[5,6] L. Castaldini,[1] A. Ciabattoni,[7,1] A. Di Piano,[1] R. Falco,[1] S. Gallego,[8] G. Mustafa,[9,1] P. Patel,[10] A. Rizzo,[11] E. A. Wulf,[12] D. H. Hartmann,[13] C. A. Kierans,[4] J. A. Tomsick,[14] and A. Zoglauer[14]

[1]*INAF/OAS Bologna, Via P. Gobetti 93/3, 40129 Bologna, Italy;* nicolo.parmiggiani@inaf.it

[2]*Department of Physics and Astronomy, Louisiana State University, Baton Rouge, LA 70803, USA*

[3]*George Washington University, 2121 I St NW, Washington, DC 20052, USA*

[4]*NASA Goddard Space Flight Center, 8800 Greenbelt Road, Greenbelt, MD 20771, USA*

[5]*Department of Astronomy, University of Maryland, College Park, MD 20742, USA*

[6]*Center for Research and Exploration in Space Science and Technology, NASA/GSFC, Greenbelt, MD 20771, USA*

[7]*Department of Physics and Astronomy "Augusto Righi", University of Bologna, Via P. Gobetti 93/2, 40129 Bologna, Italy*

[8]*Institut für Physik & Exzellenzcluster PRISMA+, Johannes Gutenberg-Universität Mainz, 55099 Mainz, Germany;*

[9]*Department of Computer Science and Engineering, University of Bologna, Via Zamboni 33, 40126 Bologna, Italy*

[10]*George Mason University, 4400 University Dr, Fairfax, VA 22030, USA;*

[11]*INAF/OA Catania, Via Santa Sofia 78, 95123 Catania, Italia*

[12]*U.S. Naval Research Laboratory, 4555 Overlook Ave., SW Washington, DC 20375, USA*

[13]*Department of Physics and Astronomy, Clemson University, Kinard Lab of Physics, Clemson, SC 29634-0978, USA*

[14]*Space Sciences Laboratory, 7 Gauss Way, University of California, Berkeley CA 94720-7450, USA*

Abstract. The Compton Spectrometer and Imager is a NASA satellite mission under development that will survey the entire sky in the 0.2-5 MeV range using a wide-field

ronomy!gamma-ray
rsts
orithm!machine
earning!deep learning
LPix! ascl:1107.018
MA! ascl:2104.024
Alib! ascl:1906.018

germanium detector array, surrounded on the sides and bottom by active shields (the Anticoincidence Subsystem, ACS). The ACS aims to suppress and monitor background events, as well as detect transient sources, such as Gamma-Ray Bursts (GRBs), through its onboard triggering algorithm. The data related to GRBs are sent to the ground and analyzed by an automated pipeline to localize the GRBs and share their positions with the community. In this work, we present a brief GRB localization method based on ACS data, utilizing deep learning (DL) techniques, which can estimate the 90% confidence region, including cases where it is split into multiple areas. To address this, we developed a neural network classifier that predicts the GRB location as a probability distribution across the sky map following the HEALPix framework. The distribution can be used to compute the 90% confidence regions. Future work will compare this DL-based localization approach with classical methods such as χ^2 fitting and Maximum Likelihood Estimation.

1. Introduction

The Compton Spectrometer and Imager (COSI, Tomsick et al. (2024)) is a NASA satellite mission under development that will survey the entire sky in the 0.2-5 MeV range with a wide-field gamma-ray telescope. Its main instrument consists of a germanium detector array, surrounded on the sides and bottom by bismuth germanate (BGO) scintillator active shields (the Anticoincidence Subsystem, ACS). The ACS both suppresses and monitors background events and enables the detection of transient sources. COSI will include an onboard triggering algorithm capable of identifying Gamma-Ray Bursts (GRBs) in the ACS and transmitting data to the ground for further analysis. The GRBs will be localized by an automated pipeline, with the localizations promptly sent to the community. The ACS is composed of 22 BGO crystals, ten on the bottom side and three on each lateral side (left panel of Figure 1). Each crystal is coupled to a 3x3 array of 6 mm Silicon Photomultipliers (SiPM), each with two gain channels: a high-gain "gamma" channel (~80 keV - 2 MeV) and a low-gain "proton" channel (>2 MeV). Both channels are read out by an Application-Specific Integrated Circuit (ASIC). Count rates are recorded every 50 ms. The analyses in this work use only the "gamma" channel. The Light Curves of the ACS segments are obtained by grouping a lateral wall per ASIC with two further ASICs reading out the bottom wall. The final two bottom BGO modules are combined with two lateral walls as illustrated in the right panel of Figure 1 to optimize the readout channels. The red boxes indicate the aggregation of bottom modules into lateral walls.

In this work, we present two short GRB (sGRB) localization methods based on ACS data and developed using deep learning (DL) techniques. Different network architectures were evaluated to estimate localization uncertainties at the 90% confidence level, including cases where the confidence region is split into multiple areas. In previous work Parmiggiani et al. (2026), we compared the localization performance of our DL model with the χ^2 fit (Connaughton et al. 2015) and the bc-tools localization methods (Martinez-Castellanos et al. 2025). However, that comparison was limited to the angular distance error between the simulated sGRB positions and the localized positions, since the DL model used in that work could not estimate the 90% confidence error regions. The DL model presented in this manuscript is capable of estimating the 90% confidence error areas, allowing for a direct comparison with classical localization methods.

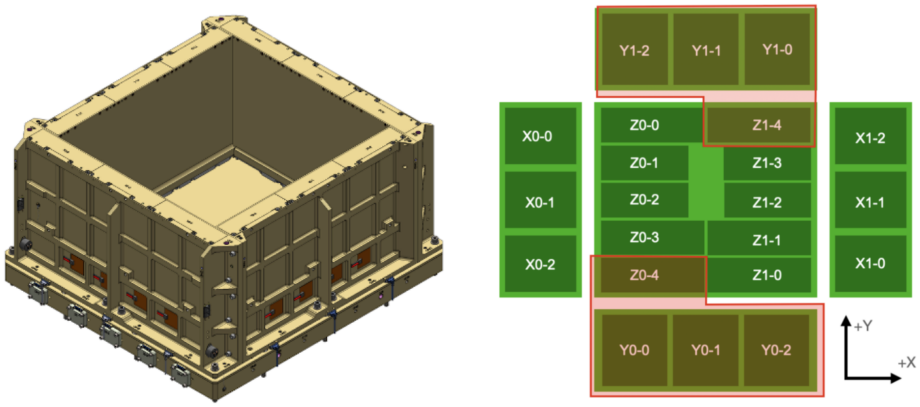


Figure 1. *Left:* COSI ACS Design. *Right:* BGO crystal scheme.

2. Methods

We simulated two sGRB datasets using MEGAlib (Zoglauer et al. 2006) for training and testing purposes. The simulation procedure is described in detail in Parmiggiani et al. (2026). The testing dataset contains 49,152 sGRBs distributed following the HEALPix coordinate system (nside=64), while the training dataset contains 100,000 sGRBs randomly sampled from the HEALPix positions with nside=128. The sGRBs are simulated with a variable flux and three different spectral models described in Parmiggiani et al. (2026).

The first DL model we developed is a probabilistic regressor designed to reconstruct the sGRB arrival direction. The model takes as input the count rates of the ACS panels and outputs a full multivariate normal probability distribution over four correlated directional components ($sin\theta$, $cos\theta$, $sin\phi$, $cos\phi$). The architecture consists of a Feedforward Neural Network with three hidden layers of 256, 128, and 64 neurons, respectively, and the LeakyReLU activation function, similar to the DL model presented in Parmiggiani et al. (2026). Dropout regularization is applied during training to reduce overfitting. The probabilistic formulation enables the model to provide not only directional estimates but also associated uncertainties.

The second DL model maps the ACS panel counts to a discrete probability distribution over sky positions represented using the HEALPix framework. The architecture of the network is identical to that of the first model except for the output layer, which uses a softmax activation function to produce a normalized probability distribution over all HEALPix pixels. The labels of the training dataset are one-hot encoded HEALPix maps representing the true source position. To improve training stability, the true positions are smoothed to avoid collapsing onto a single pixel.

3. Results

The results show that the probabilistic regression model outputs a multivariate normal distribution, describing sGRB localization uncertainties as elliptical error regions. This approach suits cases where the error distribution is unimodal and Gaussian-like. However, it cannot capture complex or multimodal error structures caused by detec-

tor geometry. In contrast, the HEALPix-based probabilistic model, which predicts a probability distribution over the sky map, enables, for the first time in our analysis with DL models, the estimation of multimodal error regions covering multiple possible sky areas, as shown in Figure 2. The HEALPix probability maps reveal richer spatial structures, capturing several peaks in the predicted localization probability.

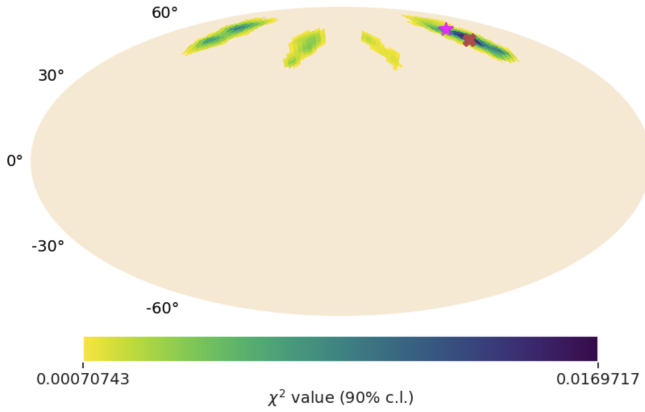


Figure 2. Examples of 90% c.l. error regions (997 deg^2) with multimodal behaviour obtained with the second DL model. The purple star marker indicates the simulated sGRB positions, while red X markers indicate the sGRB localizations.

4. Conclusion

The results of this work demonstrate that the HEALPix-based classification model can successfully estimate GRB localization error regions from COSI-ACS data, even when the probability distributions are multimodal and have multiple disjoint regions. This approach extends the model's applicability beyond simple Gaussian assumptions, providing a more realistic representation of localization uncertainty. Ongoing analyses aim to further characterize these error regions and compare them with those obtained using classical methods, such as MLE and χ^2 fitting.

Acknowledgments. The Compton Spectrometer and Imager is a NASA Explorer project led by the University of California, Berkeley with funding from NASA under contract 80GSFC21C0059. This work is supported by the Italian Space Agency, partially within contract ASI/INAF No. 2024-11-HH.0.

References

Connaughton, V., et al. 2015, The Astrophysical Journal Supplement Series, 216, 32. URL https://dx.doi.org/10.1088/0067-0049/216/2/32

Martinez-Castellanos, I., et al. 2025, bctools. URL https://doi.org/10.5281/zenodo.15419961

Parmiggiani, N., et al. 2026, The Astrophysical Journal, 997, 135. URL https://doi.org/10.3847/1538-4357/ae25fc

Tomsick, J., et al. 2024, in 38th International Cosmic Ray Conference, 745. eprint: 2308.12362

Zoglauer, A., et al. 2006, New Astronomy Reviews, 50, 629. Astronomy with Radioactivities. V, URL https://doi.org/10.1016/j.newar.2006.06.049

Astronomical Data Analysis Software and Systems XXXV
*ASP Conference Series, Vol. **VOL** Compiled: Tuesday 30th June, 2026@ 01:30*
K. Polsterer, D. Bohlender, S. Gwyn, A. Hincks and P. Teuben, eds.
*© **YEAR** Astronomical Society of the Pacific*

ZZZDEBUG!094.tex
Abu, I.|textbf
Gianotti, F.
Bulgarelli, A.
Sayes, N.
Campos, H.
Araya, M.
Borciani, E.
Bruno, P.
Calabretto, P.
Castaldini, L.
Huerta, I.
Conforti, V.
Corpora, M.
Costa, A.
Di Piano, A.
Fiordoliva, F.
Fioretti, V.
Gallozzi, S.
Giuliani, A.
Grillo, A.
Lodi, M.
Lucarelli, F.
Mandica, A.
Mattioli, G.
Munari, K.
Parmiggiani, N.
Pareschi, G.
Pastore, V.
Russo, F.
Scuderi, S.
Tacchini, A.
Tosti, G.
projects!ASTRI
databases!MongoDB
computing!cluster!Docker Swarm
packages!Docker

Development and the Deployment of the Startup System for the ASTRI Mini-Array Project

I. Abu,[a] F. Gianotti,[a] A. Bulgarelli,[a] N. Sayes,[i] H. Campos,[i] M. Araya,[i] E. Borciani,[a] P. Bruno,[b] P. Calabretto,[a] L. Castaldini,[a] I. Huerta,[i] V. Conforti,[a] M. Corpora,[e] A. Costa,[b] A. Di Piano,[a] F. Fiordoliva,[d] V. Fioretti,[a] S. Gallozzi,[d] A. Giuliani,[f] A. Grillo,[b] M. Lodi,[h] F. Lucarelli,[d] A. Mandica,[a] G. Mattioli,[a] K. Munari,[b] N. Parmiggiani,[a] G. Pareschi,[c] V. Pastore,[a] F. Russo,[a] S. Scuderi,[c] A. Tacchini,[a] G. Tosti,[g] S. Spinello[b] for the ASTRI Project[j]

[a]*INAF - Osservatorio di Astrofisica e Scienza dello Spazio, Bologna, Italy*
ismam.abu@inaf.it

[b]*INAF - Osservatorio Astrofisico di Catania, Via S. Sofia 78, Catania, Italy*

[c]*INAF - Osservatorio Astronomico di Brera, Via Bianchi 46, Merate, Italy*

[d]*INAF - Osservatorio Astronomico di Roma, Via Frascati 33, Roma, Italy*

[e]*IASF - Istituto di Astrofisica Spaziale e Fisica Cosmica di Palermo, Italy*

[f]*IASF - Istituto di Astrofisica Spaziale e Fisica Cosmica di Milano, Italy*

[g]*Università di Perugia, Dip. di Fisica e Geologia, Via A. Pascoli, Perugia, Italy*

[h]*Fundación Galileo Galilei – INAF, Santa Cruz de Gran Canaria, Spain*

[i]*Universidad Técnica Federico Santa María, Gral. Bari 699, Valparaíso, Chile*

[j]http://www.astri.inaf.it/index.php/new-general-info/new-astri-team/

Abstract. The Startup System of the ASTRI Mini-Array is a portable, containerized software that orchestrates startup and shutdown procedures and provides real-time monitoring across heterogeneous subsystems. A System Configuration Database (SCDB, MongoDB) describes the infrastructure and drives automated initialization of the telemetry and alerting backend. Admin-Tools derives from SCDB the InfluxDB layout (buckets, tasks, and conventions), while protocol-specific collectors (SNMP, OPC UA, Modbus, and Telegraf) stream measurements into InfluxDB. A lightweight web-server exposes selected streams to a operator interface and to alarm services. Development followed an incremental path from fully simulated telemetry to progressive integration of real interfaces, and deployment moved from a virtualized testbed to on-site hosts and on-sky validation with ASTRI-3. On production servers, the end-to-end stack reaches an operational ready state in about five minutes, enabling reliable recovery and consistent operations. The resulting SCDB+InfluxDB backbone is also suitable as a data foundation for future predictive maintenance and digital-twin applications.

1. Introduction

In this paper we present the Startup System, the software system responsible for managing the startup and shutdown sequence and the critical monitoring of all on-site ASTRI Mini-Array subsystems. These subsystems include Power, Environmental, Timing and ICT, Safety and Security, Service Cabinets, the Telescopes, and the SCADA system.
The ASTRI Mini-Array is an international project led by INAF. It will operate nine dual-mirror imaging atmospheric Cherenkov telescopes at the Teide Observatory in Tenerife, described in detail by Scuderi et al. (2022) and Lombardi et al. (2022b).
The array targets very-high-energy gamma-ray sources. It also enables stellar intensity interferometry, as described by Vercellone et al. (2022), Zampieri et al. (2022) and Lombardi et al. (2022a).
The observatory subsystems are coordinated by a SCADA framework, described in detail by Bulgarelli et al. (2024) and runs in the ICT System, described in detail by Gianotti et al. (2024), there is also a remote installation, in the TestBed of ASTRI Mini-Array, hosted in the INAF-OAS Bologna's Datacenter, described in detail by Gianotti et al. (2022).
The Startup System automate the operation of the SCADA Software, and it enables non-expert operators to carry out observations. Without orchestration, operators would need to manually manage dozens of interdependent services, a process prone to errors and unsafe conditions. The Startup System ensures that all critical components are started in the correct order, dependencies are respected, and the system reaches a consistent ready state.
The architecture of the Startup System is described in detail by Bulgarelli et al. (2025). This paper describes both the conceptual model and the practical path taken to realize it. We highlight the orchestration principles, the incremental development process and the results of the first deployments, including on-sky validation.

2. System Overview

The Startup System orchestrates startup and shutdown sequences and performs continuous readiness and health checks for the main on-site subsystems, including Power, Environmental, Timing and ICT, Safety and Security, Service Cabinets, Telescopes, and SCADA components. The design goal is portability: the same stack can be deployed on the on-site ICT server, an operator laptop for emergency recovery, or a remote instance hosted in the Bologna testbed and connected to the observatory through a VPN.
Deployment is intentionally simple and requires only Docker. The stack is defined as a Docker Compose application and is started through a single command interface that selects execution modes. These modes support fully simulated runs, mixed configurations, and full operation with real telemetry. The same interface provides stop procedures that either preserve databases for operational continuity or reset parts of the environment for testing and development.
Configuration is centralized in the System Configuration Database (SCDB), which provides a machine-readable description of devices, endpoints, and conventions. From SCDB, Admin-Tools generates the InfluxDB initialization (buckets, retention policies, tasks for shaping data, and naming and tagging conventions) and exports validated configuration fragments for collectors and dashboards. Collectors ingest telemetry using the relevant protocols: Telegraf agents cover host-level metrics, while dedicated ser-

vices interrogate SNMP devices, OPC UA servers, and Modbus endpoints. Measurements are stored as time series in InfluxDB, while alarms are collected and forwarded to the operator interface.

Readiness is defined by a small set of checks: service health, reachability of SCDB and InfluxDB, and the presence and freshness of key telemetry streams. Freshness checks prevent stale data from being interpreted as current status. Operational procedures are idempotent: components that are already healthy are not restarted, which reduces downtime and supports safe repeated execution during recovery and maintenance.

3. Development and Deployment

Implementation progressed in small iterations, starting from a minimal core that deployed SCDB and InfluxDB and produced synthetic telemetry. This stage validated the end-to-end data path and established the conventions used by dashboards and collectors. After stabilizing the telemetry pipeline, we introduced the real-time relay to feed the operator interface and implemented the first dashboards.

The next milestone was the alarm subsystem, which enabled fault reporting and supported operator workflows. Only after this software baseline was stable, we moved to integration with real subsystem interfaces. Protocol-specific collectors were added progressively and tested first in the Bologna testbed, which mirrors the observatory ICT environment and allows integration tests without risks for on-site operations.

As real telemetry sources were integrated, we addressed signal volume and operator usability by shaping data at the database level. InfluxDB tasks are used to filter, aggregate, and derive operational indicators while keeping raw streams available for engineering analyses. This approach reduces code churn because changes in the displayed or alarmed signals can be implemented by updating tasks rather than modifying collectors.

Deployment followed three steps: (i) testbed validation of the conceptual model, (ii) on-site installation with simulated telemetry to verify packaging and runtime behavior, and (iii) on-site operation with real telemetry and SCADA interactions to support observation procedures.

4. Validation and Test

Validation combined integration testing, fault injection, and performance measurements. Functional tests verified that the system reaches a ready state when the defined checks are satisfied and that repeated startup or shutdown executions are safe due to idempotent procedures.

Fault scenarios were exercised by simulating failures along subsystem chains and by inducing resource and network degradation. In these conditions, dashboards highlighted degraded components and alarms were produced with meaningful severity, enabling operators to identify and react to problems quickly.

Performance was measured on representative hosts. A full installation from scratch reaches an operational ready state in about five minutes on a production-class server, and within about eight to ten minutes on a laptop-class machine. Finally, the stack was validated during real observation tests, where it operated with real telemetry and interacted correctly with SCADA components during on-sky activities with ASTRI-3.

5. Conclusions and future work

The Startup System provides a portable and easy-to-install software solution that automates startup and shutdown operations and supports continuous monitoring for the ASTRI Mini-Array Project. The deployment strategy is flexible and includes a standard on-site server, an operator laptop for emergency control, and a remote recovery installation hosted in Bologna and connected to the observatory through a dedicated VPN.

The system relies on Docker as its only external dependency and uses a Docker Compose stack that can be started through a dedicated run command supporting multiple execution modes. The architecture combines telemetry ingestion into InfluxDB, alarm collection through a dedicated alarm server, and a webhook-based delivery mechanism that keeps the frontend updated. A key element is the System Configuration Database, which stores a detailed description of the observatory and enables consistent behavior across collectors, dashboards, and orchestration logic. Readiness checks and safe repeated execution of procedures improve reliability and operational safety, and they also simplify development and maintenance.

A major future direction is the use of SCDB and InfluxDB as the base for a Digital Twin. SCDB describes the structure and configuration of the system, while InfluxDB provides the time series history of subsystem behavior. Together they can support simulation, prediction, and what-if analyses. This approach is also relevant for future adoption in the SST context for CTAO, where a similar orchestration and monitoring layer could benefit from the same data-driven foundation.

Acknowledgments. This work has been carried out in the framework of the ASTRI Mini-Array Project, led by the Italian National Institute for Astrophysics (INAF). The project is supported by the Italian Ministry of University and Research (MUR) and the Ministry for Economic Development (MISE). We gratefully acknowledge the Instituto de Astrofísica de Canarias (IAC) for hosting the ASTRI Mini-Array at the Teide Observatory. Additional support has been provided by international funding agencies including FAPESP (Brazil) and the South African Department of Science and Technology. The authors also acknowledge the contributions of the ASTRI team members involved in the design, implementation, and integration of the software and ICT infrastructures.

References

Bulgarelli, A., Abu, I., & et al. 2025, in Proceedings of ADASS, vol. This Conference

Bulgarelli, A., Lucarelli, F., Tosti, G., Conforti, V., Parmiggiani, N., Schwarz, J. H., Araya, M., & et al. 2024, Journal of Astronomical Telescopes, Instruments, and Systems, 10

Gianotti, F., Abu, I., Lodi, M., Tacchini, A., & et al. 2024, in Proceedings of SPIE, vol. 13101

Gianotti, F., Conforti, V., Tacchini, A., Bulgarelli, A., Bruno, P., & et al. 2022, in Astronomical Data Analysis Software and Systems XXX, ASP Conference Series, vol. 532

Lombardi, S., Lucarelli, F., Bigongiari, C., & et al. 2022a, in Proceedings of SPIE, vol. 12189

Lombardi, S., Scuderi, S., Gargano, C., Giordano, V., Giuliani, A., Grivel, C., La Palombara, N., Lessio, L., Millul, R., & et al. 2022b, in Proceedings of SPIE, vol. 12189

Scuderi, S., Giuliani, A., Pareschi, G., Tosti, G., Catalano, O., Amato, E., Antonelli, L. A., Becerra Gonzáles, J., & et al. 2022, Journal of High Energy Astrophysics, 35

Vercellone, S., Bigongiari, A., Burtvoi, M., Cardillo, M., Catalano, O., & et al. 2022, in Journal of High Energy Astrophysics, Volume 35, August 2022, Pages 1-42, vol. 12189

Zampieri, L., Bonanno, G., Bruno, P., & et al. 2022, in Proceedings of SPIE, vol. 12183

Astronomical Data Analysis Software and Systems XXXV
*ASP Conference Series, Vol. **VOL** Compiled: Tuesday 30th June, 2026@ 01:30*
K. Polsterer, D. Bohlender, S. Gwyn, A. Hincks and P. Teuben, eds.
*© **YEAR** Astronomical Society of the Pacific*

ZZZDEBUG!095.tex
Villa, F.|textbf
Fernández, M.
Siles, A.
Espinosa, J.
Belanger, G.
Kajava, J.
Blay, P.
FOOBAR!conference!ADASS 2019
FOOBAR!organisations!ASP
FOOBAR! ascl:1101.010

Integral Science Legacy Archive: Official Release and Modular Innovation for ESA Science Archives

Fernando Villa,[1] Mónica Fernández,[1] Antonio Siles,[1] Javier Espinosa,[1] Guillaume Belanger,[2] Jari Kajava,[3] and Pedro Blay.[3]

[1]*Starion for ESA, Villafranca del Castillo, Madrid, Spain;*
fernando.villa@ext.esa.int monica.fernandez.barreiro@ext.esa.int
antonio.silesaguilar@ext.esa.int javier.espinosa@ext.esa.int

[2]*ESA(ESAC), Villafranca del Castillo, Madrid, Spain;*
guillaume.belanger@esa.int

[3]*Telespazio for ESA, Villafranca del Castillo, Madrid, Spain;*
jari.kajava@ext.esa.int pedro.blay@ext.esa.int

Abstract. The Science Archives, managed by the ESAC Science Data Centre (ESDC), serve as the definitive repositories for ESA mission data, safeguarding scientific knowledge and ensuring global accessibility for the research community.

The year 2025 marked a major milestone: the publication of the first operational release of the Integral Science Legacy Archive (ESDC Team, ESA/ESAC 2025b), now established as the official archive for INTEGRAL mission data. Developed in close collaboration with mission scientists, ISLA delivers a modern, user-centric experience, transitioning from conventional data access to specialized science portals.

ISLA is powered by web frameworks such as Angular and a suite of reusable components, together with a backend built on Spring Boot. The integration of the ESASky portal, enables exploration of high-energy sources, exposure maps, GRB catalogs, related publications, and catalog exploration. ISLA also provides a direct link from its data search interface to ESA Datalabs, where data can be processed without the need to install software or download data.

Importantly, ISLA has served as a precursor for many of the concepts and reusable widgets now adopted across other ESA Science Archives, while also benefiting from shared components developed in parallel by other teams. This collaborative and modular approach accelerates the integration of new features, and supports rapid development and future scalability.

1. Introduction

The International Gamma-Ray Astrophysics Laboratory (INTEGRAL) observed the high-energy Universe for more than two decades, from its launch in October 2002 until the end of operations in February 2025. Over its lifetime, INTEGRAL delivered transformative results in time-domain and multi-messenger astronomy. The Integral Science Legacy Archive (ISLA) (ESDC Team, ESA/ESAC 2025b) has been established at ESAC as the official archive to provide long-term access to the consolidated mission data and high-level science products.

ISLA embodies the evolution of ESA's science archives toward user-centric science portals, complementing classical data forms with rich discovery and analysis capabilities. The first public release, announced in March 2025, integrates exploration tools, cross-mission discovery via ESASky (Baines et al. 2017)(Giordano et al. 2018), community contributions, and seamless links to cloud-based data exploitation platforms.

2. Background and Motivation

INTEGRAL was the first space observatory to simultaneously observe celestial objects in gamma rays, X-rays, and visible light. It permitted the discovery of rare transient events. Since launch, the INTEGRAL Science Data Centre (ISDC) was responsible for the analysis and processing of INTEGRAL data, which are made directly usable by scientists.

With the end of observations in February 2025, INTEGRAL data transitioned to ESAC (Figure 1), through ISLA. The ESAC Science Data Centre (ESDC Team, ESA/ESAC 2025a) located at the European Space Astronomy Centre (ESAC) in Madrid, Spain, provides services and tools to access and retrieve observations and data from ESA's space science missions, in areas of astronomy, planetary science and heliophysics.

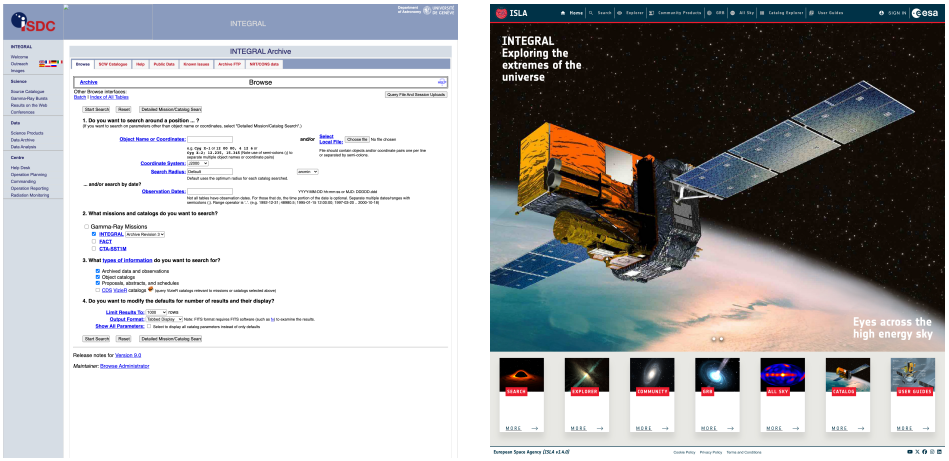


Figure 1. From ISDC to ESAC. *Left:* INTEGRAL Archive (ISDC). *Right:* ISLA Archive (ESAC).

ISLA provides access to all consolidated data as well as high level products like time series, spectra and mosaics per observation for the high energy sources.

3. System Architecture

In the specific case of ISLA, significant efforts were made to involve Mission Scientists. The goal was to enhance the user experience by transitioning from traditional data forms to specialized Scientific Portals (Figure 2).

In the Search Portal, there are two methods for querying mission observations: One based on a sophisticated grammar, and another that uses an auto-complete search

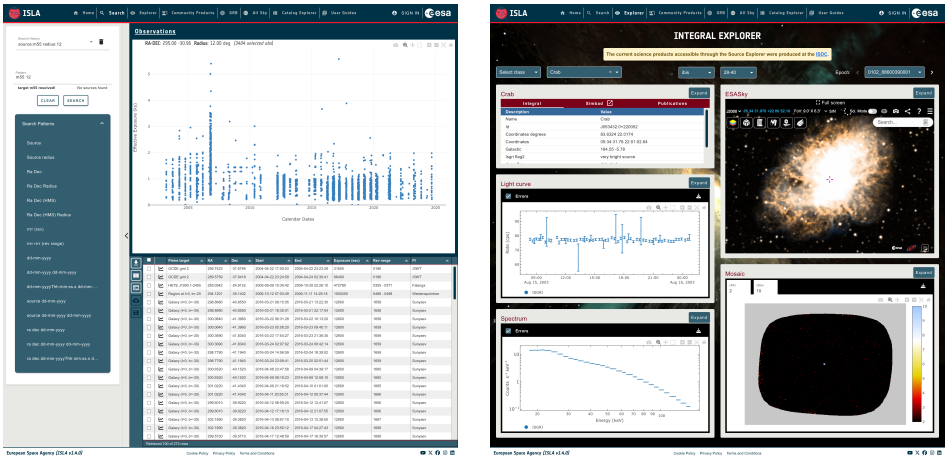


Figure 2. *Left:* ISLA Search Portal with advance grammar, observations results and light curves. *Right:* Explorer Portal showing MMODA products from ISDC.

feature. This search tool leverages both the high-energy INTEGRAL catalog and SIMBAD to locate the desired celestial objects, displaying the associated light curves and relevant metadata.

A dedicated portal (Explorer Portal) has been created to explore MMODA products (Multi-Messenger Online Data Analysis)(ISDC Team 2025). It is a platform based at the University of Geneve that uses complex pipelines to generate high-level scientific products. In this Portal, the user selects a source and four interactive graph panels and a metadata table are displayed. The metadata table shows all the information about the high-energy source stored in the Archive and SIMBAD databases, and the associated publications.

ISLA also integrates ESASky (Baines et al. 2017)(Giordano et al. 2018), an application that allows to visualize and download public astronomical data. It was integrated into different Portals: Explorer Portal, GRB Portal (Gamma-ray Bursts), and All Sky Portal (IBIS Exposure Maps). Another key widget is the ESDC Table, which supports multi-row selection, sending data to ESA Datalabs (ESDC: ESAC Science Data Centre 2025), downloading metadata, and choosing visible columns.

All this flexibility is enabled by adopting Angular as the common frontend framework across the ESA archives ecosystem, which provides a suite of reusable widgets such as cards, image viewers, and splash screens.

The backend is also built with modular Java services (Spring Boot) that implement IVOA Table Access Protocol (TAP) (International Virtual Observatory Alliance 2025), and is registered in the Euro-VO Registry (Euro-VO Registry 2025) for discoverability. PyVO provides programmatic Python access for automated and reproducible analyses.

4. Discussion: Modularity, Legacy and Roadmap

Modularity fosters innovation and scalability. By encapsulating UI widgets and backend services. ISLA's architecture supports rapid feature integration, reduces maintenance cost, and facilitates future evolution. Registration in VO registries, adherence to IVOA standards, and integration with ESASky strengthen discoverability and cross-

mission analysis. The community products mechanism expands the legacy beyond core instrument pipelines, capturing domain knowledge and value-added results for future generations.

All these core concepts have also influenced and have been applied to other ESDC archives like Hubble (ESDC Team, ESA/ESDC, 2025), James Webb (ESDC Team, ESA/ESAC 2025c) and XMM-Newton (ESDC Team, ESA/ESAC 2025d).

Near-term enhancements include an ADQL console for browser-based querying. The progressive migration of reusable components across ESA archives will harmonize user interfaces and support integrated cross-archive scientific workflows.

5. Conclusions

ISLA first operational release establishes the official legacy archive for INTEGRAL, delivering a modern science portal experience and VO-compliant services. Its modular approach—from Angular widgets to TAP services—is propagating across ESA's archive ecosystem, consolidating best practices and accelerating innovation.

Acknowledgments. We thank the INTEGRAL mission teams, the ISDC, and the ESAC Science Data Centre for their support. We also acknowledge the ESA archives community, the ESASky team, and the Astroquery team for their contributions.

References

Baines, D., et al. 2017, Publications of the Astronomical Society of the Pacific, 129, 028001. Complementary reference to Giordano et al. (2018) for ESASky architecture and design

ESDC: ESAC Science Data Centre 2025, ESA Datalabs: Cloud-Based Science Platform for ESA Archives, `https://datalabs.esa.int/`. ESA Datalabs official platform for data exploitation

ESDC Team, ESA/ESAC 2025a, Esdc: Esac science data centre, `https://www.cosmos.esa.int/web/esdc`. ESA Science Data Centre: hosting and maintaining ESA Science Archives and Virtual Observatory services

— 2025b, Isla: Integral science legacy archive, `https://isla.esac.esa.int/`. ESA Science Archives official release

— 2025c, James webb space telescope (jwst) science archive, `https://jwst.esac.esa.int/ejwst`. ESA Science Archive portal for JWST mission data

— 2025d, Xmm-newton science archive (nxsa) gallery, `https://nxsa.esac.esa.int/nxsa-web/#gallery`. ESA Science Archive portal for XMM-Newton mission data and gallery

ESDC Team, ESA/ESDC, 2025, Hubble space telescope (hst) science archive, `https://hst.esac.esa.int`. ESA Science Archive portal for HST mission data

Euro-VO Registry 2025, Euro-VO Resource Registry, `https://registry.euro-vo.org/`. Harvestable VO resource registry

Giordano, F., et al. 2018, Astronomy and Computing, 24, 97. See also Baines, D. et al. (2017), PASP, 129, 028001: https://ui.adsabs.harvard.edu/abs/2017PASP..129b8001B/abstract

International Virtual Observatory Alliance 2025, IVOA Standards and Protocols, `https://www.ivoa.net/`. Virtual Observatory standards for interoperability

ISDC Team 2025, INTEGRAL MMODA Products, `https://www.astro.unige.ch/astroordas/mmoda-product-gallery`. Multi-Messenger Online Data Analysis Products

Astronomical Data Analysis Software and Systems XXXV
*ASP Conference Series, Vol. **VOL** Compiled: Tuesday 30th June, 2026@ 01:30*
K. Polsterer, D. Bohlender, S. Gwyn, A. Hincks and P. Teuben, eds.
*© **YEAR** Astronomical Society of the Pacific*

ZZZDEBUG!096.tex
Siejkowski, H.|textbf
Balawajder, T.
Leśniak, M.
Kocot, J.
Milian, K.
Sto, J.
Wenda, B.
software!open source
software!user interfaces!web-based
web!development tools!Google Web Toolkit (GWT)
web!application
computing!high-performance computing (HPC)
software!workflows
FOOBAR! ascl:1101.010

Practical lessons in building collaborative science platforms with open-source tools

Hubert Siejkowski, Tomasz Balawajder, Maciej Leśniak, Joanna Kocot, Krystyna Milian, Jakub Sto, and Bartłomiej Wenda

Academic Computer Centre CYFRONET of the AGH University of Krakow, Krakow, Poland; hubert.siejkowski@cyfronet.pl

Abstract.

Advanced science platforms must handle large data volumes, complex workflows, and collaborations that span multiple disciplines and partners. While scientific questions may differ between fields, the challenges of building reliable and reproducible data-driven research infrastructures are very similar.

We show how a geophysics-oriented science platform integrates well known open-source services to support the full lifecycle of data-intensive research. The platform consists of monitoring, log aggregation and analysis tools. These tools provide real-time insights into the health and efficiency of the platform and scientific workflows. They are also generating metrics and usage reports that satisfy accountability requirements of funding agencies.

Beyond infrastructure monitoring, collaboration and reproducibility are enabled through distributed version control, allowing users to develop and run custom applications on our platform. To support the increasing role of machine learning in scientific workflows, we integrate tools for experiment tracking, model management and collaborative dataset curation and annotation. The platform supports execution on remote workers and HPC clusters, enabling scalable computation and flexible integration with existing research infrastructures.

Our experience demonstrates that combining these tools into a modular ecosystem creates science platforms that are scalable, and cultivate reproducibility. Key lessons stress the need to balance automation with researcher control, design platforms that enable cross-disciplinary collaboration, and adopt testing frameworks to ensure platform reliability. Although developed for geophysics, this approach is directly transferable to astrophysics and other data-intensive fields.

1. Introduction

The EPISODES Platform[1] is a collaborative research platform for conducting research in geophysics (Orlecka-Sikora et al. 2020). The platform focuses on anthropogenic seismology, which deals with seismic activity caused by human activity, such as mining or reservoir impoundment. The development of the platform is a collaboration between ACC Cyfronet AGH and the Institute of Geophysics Polish Academy of Sciences, combining expertise in IT and geophysics, respectively.

[1] https://EpisodesPlatform.eu

The EPISODES Platform is one of the Thematic Core Services with the European Plate Observing System[2] (EPOS). EPOS is a European Research Infrastructure Consortium (ERIC) that serves as a multidisciplinary, distributed network. Its purpose is to facilitate the integrated use of data, data products, and facilities from the solid Earth science community across Europe.

Target users are students, lecturers, and entry- and mid-level scientists. The main functionality of the EPISODES Platform is to simplify data selection and perform subsequent analysis steps. Users can create workflows and reuse them in the future, allowing for a streamlined analysis process. Workflows can be shared between users and developed jointly. Users can also develop their own applications and use them in data analysis or even share them with the entire user community.

2. Aims of the EPISODES Platform

The primary objectives of the EPISODES Platform are to enhance research in anthropogenic seismology by providing an integrated environment for data analysis, workflow creation, and resource sharing. The typical user scenario using the platform is as follows:

Find and select data. Data storage systems are distributed and hosted by different institutions. The EPISODES Platform provides a frontend for accessing these systems, allowing users to browse the data, select it, and attach it to their user workspace. The download process runs behind the scenes. Each data set is assigned a data type, which is helpful in later steps.

Run applications. The applications can be run using the downloaded data as an input. The user can upload their own data as well. The assigned data types (described in the previous step) are used to limit the selection of applications suggested by the system, since each data type has a list of applications that can process it.

Create custom applications. In the case if the selection of prebuilt applications is too restrictive, the user can create their own applications and execute them on the platform. A custom application must provide the list of data types that it can process and the list of outputs (including their data types). The development of a custom application can be done on the platform. It is possible to develop a custom application in a local development environment and then upload any working version to the platform.

Join application into workflows. Individual applications can be organized into workflows that can combine each analysis step into a single executable workflow. The workflows take a list of inputs that represents the combined set of inputs required by each application within the workflow. The outputs comprise all of the outputs generated by each application.

Collaboration. Each folder containing applications, together with its inputs and outputs, as well as the workflows, can be shared with a selected group of collaborators. The shared resources can be further developed by those users.

[2] https://www.epos-eu.org

3. Internals of the EPISODES Platform

The platform consists of a distributed set of tools that are connected to the central web portal. The web portal, which is the heart of the system, is a Java application built with the Google Web Toolkit[3] (GWT). This portal hosts the user workspace and governs the computations. The persistence of the system is supported by a MongoDB database (Sholmes et al. 2023) which stores information about the user profile and workspace. The data files used by the users in their analysis are kept on the local filesystem; files exceeding a size threshold are sent to an object storage backed by the S3 Ceph Object Gateway service.

Authorization and authentication (AA) are crucial due to the distributed nature of the EPISODES Platform. The service driving our AA is Unity IdM[4], which implements the OpenID Connect standard (Gupta 2024). This enables users to log into all tools using a single set of credentials, eliminating repeated logins and password fatigue in a Single Sign-On (SSO) model.

One of the core functionalities of the platform is to run applications that perform scientific analysis tasks. To move the computations off the platform, we introduced a system of workers that are responsible for fetching the input data, performing the computations, and sending back the results. The workers are Java services that can be run on any machine but require a connection to the task queue. The task queue is managed by an ActiveMQ instance (Snyder et al. 2011). Tasks can differ in their input size and the required computing resources; therefore, the platform supports two types of tasks: simple and intensive. The simple tasks are routed to workers that are constatntly online, and they process the arriving tasks almost instantly. More intensive tasks are run on a remote HPC cluster. Connection to the cluster is via the SSH gateway, which allows the platform to submit SLURM jobs (Yoo et al. 2003). A worker is started within the SLURM job and processes tasks from the task queue. In this setup, the worker can run computationally intensive tasks, which require e.g. multiple CPU cores.

The applications run by the workers are predefined in the EPISODES Platform; however, users can also create their own custom applications. The Application Workbench is an instance of Gitea (a Git hosting platform) integrated into the EPISODES Platform. The workers download the custom application automatically via the REST API, without user interaction. The only requirement on the worker side is the local installation of the necessary software or library dependencies. Workers support running applications written in Python, R, Octave, Matlab, or bash scripts.

For the Python custom applications, environment requirements can be defined in the source code repository, as the worker has built-in support for the Micromamba environments. In this case, the worker fetches the source code and creates the Python environment based on the provided definition before running the application. This allows users to easily incorporate any libraries required to perform their analysis tasks.

From the perspective of platform maintainers and managers, tools have been developed to monitor both the current status and health of the system and measure and collect usage metrics. The metrics are used in reports to stakeholders, demonstrating that the EPISODES Platform has a growing user base and is actively used by the community.

[3] `https://www.gwtproject.org`

[4] `https://unity-idm.eu`

From the developers' point of view, the metrics are very helpful for understanding how users interact with the platform and for identifying potential problems. For example, the metrics are used to estimate the runtime of a given application to determine whether it should be submitted to the HPC cluster because its potential runtime is too long for the workers assigned to the simple tasks.

The collection of the metrics from the systems is performed with the aid of Prometheus and Flunetd. The first tool is responsible for collecting the metrics from monitoring agents, while the second collects log entries from multiple sources and aggregates them into a single database. To explore the collected monitoring entries, we use Grafana. Log entries are aggregated into a database powered by ElasticSearch with a user-friendly front end powered by Kibana. On a daily basis, the collected log entries are scanned for potential errors. If any are detected, an alert is sent to the monitoring team. The alerting system is based on Alertmanager and ElastAlert.

4. Summary

From more than ten years of experience developing the platform, we have identified the following key points: (1) The complex research platform for a specific domain can be implemented using open-source tools. Building on open-source tools and standards ensures interoperability, eases maintenance, and prevents vendor lock-in, which is critical for long-term research infrastructure; (2) Many open-source solutions need only slight modifications to meet the requirements of scientific research problems, and additional integration logic is necessary to connect these tools into a unified platform; (3) Implementing a robust Single Sign-On system is crucial; (4) Close collaboration between developers and researchers with expertise in the domain is essential; (5) Comprehensive monitoring of the system provides significant benefits for both users and platform managers; (6) The architecture of such a platform must be designed for adaptability and be ready to meet future demands.

Acknowledgments. This work was conducted in the framework of the project co-financed by the Minister of Science under contract no. 2024/WK/05. With support from the EPOS ON project (G.A. n10113159) co-financed by the Polish Ministry of Science and Higher Education within programme for International Co-financed Projects and EC Horizon Europe programme. We gratefully acknowledge support Polish high-performance computing infrastructure PLGrid (HPC Center: ACK Cyfronet AGH) for providing computer facilities and support within computational grant no. PLG/2025/018833 and PLG/2025/01884.

References

Gupta, A. 2024, IDPro Body of Knowledge, 1. URL `https://doi.org/10.55621/idpro.109`

Orlecka-Sikora, B., Lasocki, S., Kocot, J., Szepieniec, T., Grasso, J. R., Garcia-Aristizabal, A., Schaming, M., Urban, P., Jones, G., Stimpson, I., & et al. 2020, Scientific Data, 7, 89

Sholmes, E., Chodorow, K., & DiGiano, D. 2023, MongoDB: The Definitive Guide (Sebastopol, CA: O'Reilly Media), 3rd ed.

Snyder, B., Bosanac, D., & Davies, R. 2011, ActiveMQ in Action (Manning)

Yoo, A. B., Jette, M. A., & Grondona, M. 2003, in Job Scheduling Strategies for Parallel Processing, edited by D. Feitelson, L. Rudolph, & U. Schwiegelshohn (Berlin, Heidelberg: Springer Berlin Heidelberg), 44

Astronomical Data Analysis Software and Systems XXXV
*ASP Conference Series, Vol. **VOL** Compiled: Tuesday 30th June, 2026@ 01:30*
K. Polsterer, D. Bohlender, S. Gwyn, A. Hincks and P. Teuben, eds.
*© **YEAR** Astronomical Society of the Pacific*

ZZZDEBUG!097.tex
Hunt, E. L.|textbf
Carpenter, V. S.
Cook, K. W.
Hilton, D. G.
Asad, M.
Jochum, J.
Lepo, K.
Zvirzdin, J.
web!social media
computer
languages!Python
FOOBAR! ascl:1101.010

The Astrosky Ecosystem: An independent online platform for science communication and social networking

Emily L. Hunt,[1] Vincent S. Carpenter,[2] Kyle W. Cook,[3] Douglas G. Hilton,[4] Mehnaaz Asad,[2] Janine Jochum,[1] Kelly Lepo,[5] and Jamie Zvirzdin[6]

[1]*Department of Astrophysics, University of Vienna, Türkenschanzstrasse 17, 1180 Wien, Austria;* emily.lauren.hunt@univie.ac.at
[2]*Contributor, The Astrosky Ecosystem*
[3]*Department of Physics & Astronomy, University of Louisville, Natural Sciences Building 102, Louisville, KY 40208, USA*
[4]*School of Earth and Space Exploration, Arizona State University, Tempe, AZ 85287, USA*
[5]*Space Telescope Science Institute, 3700 San Martin Drive, Baltimore, MD 21218, USA*
[6]*High-Energy Astrophysics Institute, Department of Physics & Astronomy, University of Utah, Salt Lake City, UT 84112, USA*

Abstract. While almost everything that astronomers study occurs in the vacuum of space, astronomy itself does not 'happen in a vacuum'. Interactions between scientists, as well as outreach to members of the public, improve extensively from access to good communication tools. Social media has become a key tool for communication in astronomy, being widely used by individuals and organizations alike for networking, outreach, and more. However, traditional social media is reliant on benevolent corporations providing a free service without compromising on quality, and the recent takeover and decline of Twitter has shown how vulnerable these platforms can be. In this proceeding, we present The Astrosky Ecosystem, which is an initiative to develop open-source tools and integrations for social media, principally the Bluesky social network. We explain how our project enables the astronomy community to operate its own social media infrastructure, independent of for-profit corporations. We also discuss some of the project's technical aspects, including its use of the AT Protocol for social networking, before concluding with ideas for the future.

1. Introduction

Social media has revolutionized communication in astronomy. For outreach, social media offers an incredible opportunity to have fast, two-way communication with broad audiences. Varied uses of social media range in size from large organizations to individuals posting about their own work (Cominsky et al. 2014; Blanton et al. 2017; Russo et al. 2019; Dalgleish et al. 2022; Smethurst 2024). For researchers, social media platforms offer a wide range of benefits, including as a tool for networking, discussing papers, or following conference news (Shuai et al. 2012; Darling et al. 2013; Brooks et al. 2020; Woods & Walker 2022).

Owing to its real-time 'microblogging' format, Twitter emerged as the main platform for discourse and outreach in astronomy. However, as 'products' ran on a for-profit basis, traditional social media platforms are vulnerable to the whims of their owners. This fate unfortunately befell Twitter (now known as X) in 2022. Since then, X has plummeted in popularity across science because of a multitude of factors, including a sharp rise in hate speech and disinformation (Mallapaty 2024; Kupferschmidt 2024). The weakness of social media tools stands in stark contrast to the rest of the astronomical ecosystem: almost all other tools (such as the arXiv or Python packages) do not just disappear overnight. However, this is the current reality of the tools used for communication in astronomy, and the gradual decline of X has caused more than a decade of online astronomy community building to be lost.

In this work, we offer an alternative. Instead of relying on the whims of for-profit social media, we show that new social media technology can allow our community to host its own open-source social media infrastructure on a small budget. We discuss the Authenticated Transfer (AT) Protocol and how it enables this. Additionally, we present our experiences from the first 2.5 years of running The Astrosky Ecosystem online.

2. The AT Protocol

The core problem of running a social network is the need to ingest and distribute large amounts of data in real time. Traditionally, this has limited operating social networks to large corporations, who usually build bespoke, non-interoperable services that must be run on a for-profit basis to be financially sustainable. In this model, the corporation owning the platform has complete control over all content and data on it and can even sway algorithms when and how they please – such as amplifying disinformation if it increases user engagement and hence advertising revenue (Aïmeur et al. 2023).

Modern, decentralized, and highly interoperable social media protocols like the AT Protocol offer an alternative to this model. The AT Protocol (Kleppmann et al. 2024) is an open-source protocol that splits traditional social media into multiple separate components. All data for a user across all AT Protocol applications is hosted on a Personal Data Server (PDS), with data types (such as posts on an app) following publishable schemas. Data from all PDSes is aggregated by a relay. Relays then distribute this data to other services, including moderation services, feed generation services, or applications. Finally, applications show a view of the network to users, and their interactions with it (such as 'likes') are stored on their PDS and distributed around once again. All of these components can be hosted independently by any entity, and by virtue of extensive protocol design and optimization efforts, all components remain interoperable and cost-effective at scale (Kleppmann et al. 2024).

The most prolific example of an AT Protocol app is Bluesky. Bluesky is an open-source Twitter-like 'microblogging' platform that has risen to prominence within science, becoming one of the de-facto replacements for Twitter (Fieldhouse 2025). By virtue of being an AT Protocol-based app, components of the platform (such as feed generators or PDS hosting) can already be run independently. In the event that the company operating Bluesky becomes an adversary, users would be able to migrate their data to new PDS hosts and pick up where they left off on a new clone of the original Bluesky app. Amongst a lively open-source AT Protocol developer community, multiple parties are already working on ensuring that this resilience exists within the network, such as the Blacksky and Eurosky initiatives.

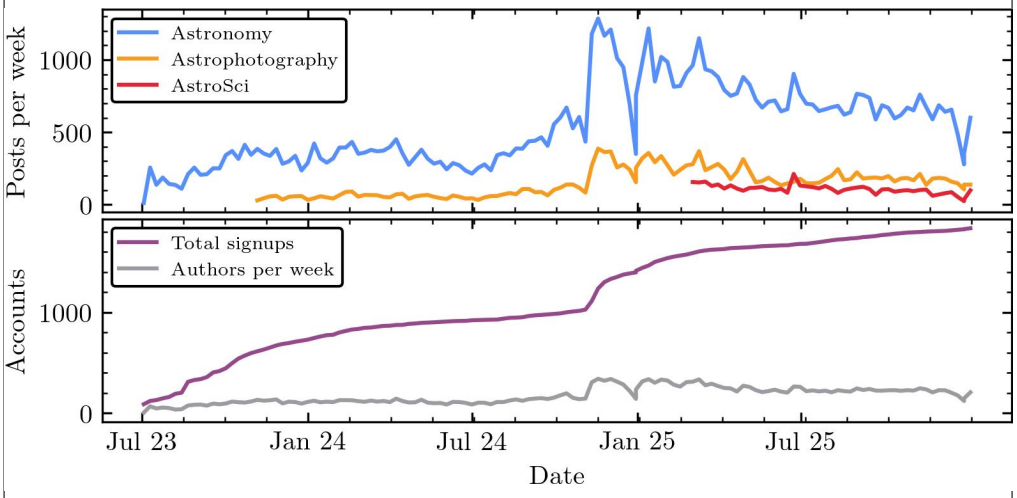


Figure 1. Top: Number of posts per week to the Astronomy, Astrophotography, and AstroSci feeds made by registered posters. Bottom: Total number of registered posters on the feeds and the number of unique post authors per week.

3. The Astrosky Ecosystem

Beginning in July 2023, our project has been curating an ecosystem of open-source tools[1] within the AT Protocol space to empower astronomers and astronomy organizations online. We aim to not only replicate the mostly positive 2010s experience of Twitter for astronomy but even exceed it. Our main effort so far has been in producing and curating the Astronomy feeds, which are a network of feeds with posts about astronomy and astrophotography that can be viewed on Bluesky. In total, we operate seventeen chronological feeds of posts, the three most popular of which currently see hundreds of posts per week from around 200–300 different users (Figure 1).

The feeds have formed a central meeting point for the astronomy community and astronomy enthusiasts on Bluesky. Some of the largest accounts posting to the feeds include the European Space Agency, the Rubin Observatory, the European Southern Observatory, and the American Astronomical Society. In addition, the feeds include a diverse community of individual astronomers – of which many registered posters are astrophotographers, who help the public engage with astronomy and whose eye-catching posts show that the night sky is observable by anyone (Sparks & Kruse 2019).

Since March 2025, we have also been logging the number of requests to our webserver. Our server has seen an average of 740 000 requests per week across the last ten months from an average of 92 000 different accounts per week. Another metric for feed interaction is the number of times someone 'scrolls' the feed (i.e., views more than the first ~20 posts): On average, the feeds are scrolled 91 000 times per week by 19 000 unique accounts. Posting to the feeds can massively elevate the reach of posts by astronomers and astrophotographers online, with even small accounts able to reach a wide audience. The number of viewers of the feeds far exceeds the total number of registered posters (1836 as of January 2026).

[1] https://github.com/the-astrosky-ecosystem

These feeds, and all associated data to run them, are hosted independently by our project. All source code for the feeds is written in Python, chosen because of its familiarity to most astronomers, and the source code is available as an open-source library on GitHub. In addition, the feeds are moderated to be free of disinformation, including a sign-up step before users can post to the feeds. This ensures that only real astronomers or astrophotographers post to the feeds. We recently began a recurring crowdfunding campaign, which now financially covers our hosting costs (approximately 40 euro per month) in addition to some other project expenses.

In the near future, we are planning on expanding the range of the different tools we offer. Our next priority will be offering PDS (user data) hosting for astronomers online, cementing the independence and longevity of all astronomy content on AT Protocol applications. Better integrations with conferences, the arXiv, or telescopes are a possibility, particularly given the option for AT Protocol to quickly build new applications or data schemas while reusing existing account and data infrastructure. We are also investigating ways to further improve our feeds, such as by offering options to partially filter content by a user's primary interests or improving our registration process.

It is clear that the real-time, independent, and open-source functionality of the AT Protocol offers many advantages for social networking for astronomy. Many avenues to apply this new technology remain unexplored, and many more ideas could be explored in the future: for example, integrating the AT Protocol with existing publication infrastructure or even broadcasting detected astronomical transients over the AT Protocol. We welcome new contributions to our project, and we are excited to see where the AT Protocol may lead as we share and discuss astronomy online.

Acknowledgments. We thank previous moderators and developers for their contributions to the project. We also thank the many developers in the lively and friendly open-source AT Protocol community for their support and encouragement. In particular, we thank Ilya Siamionau for developing the AT Protocol Python SDK, and for his help with bug fixes throughout the past 30 months. The authors thank Open Collective Europe for financially hosting the project.

References

Aïmeur, E., Amri, S., & Brassard, G. 2023, Social Network Analysis and Mining, 13, 30
Blanton, M. R., et al. 2017, The Astronomical Journal, 154, 28
Brooks, B., Brooks, K., Hagen, L., Hathi, N., Hoffman, S., Paranilam, J., & Prichard, L. 2020, arXiv e-prints, arXiv:2007.10970
Cominsky, L., McLin, K., Simonnet, A., & the Swift E/PO Team 2014, arXiv e-prints, arXiv:1405.2104
Dalgleish, H., et al. 2022, in 37th International Cosmic Ray Conference, 1397
Darling, E. S., Shiffman, D., Côté, I. M., & Drew, J. A. 2013, arXiv e-prints, arXiv:1305.0435
Fieldhouse, R. 2025, Nature, 645, 295
Kleppmann, M., et al. 2024, arXiv e-prints, arXiv:2402.03239
Kupferschmidt, K. 2024, Science, 386, 950
Mallapaty, S. 2024, Nature, 636, 15
Russo, P., et al. 2019, arXiv e-prints, arXiv:1908.01546
Shuai, X., Pepe, A., & Bollen, J. 2012, PLoS ONE, 7, e47523
Smethurst, R. 2024, in EAS2024, European Astronomical Society Annual Meeting, 242
Sparks, R. T., & Kruse, B. 2019, in Advancing Astronomy for All: ASP 2018, edited by G. Schultz, J. Barnes, & L. Shore, vol. 524 of ASP Conf. Ser., 115
Woods, P., & Walker, A. 2022, Nature Astronomy, 6, 622

Astronomical Data Analysis Software and Systems XXXV
*ASP Conference Series, Vol. **VOL** Compiled: Tuesday 30th June, 2026@ 01:30*
K. Polsterer, D. Bohlender, S. Gwyn, A. Hincks and P. Teuben, eds.
*© **YEAR** Astronomical Society of the Pacific*

ZZZDEBUG!102.tex
Holties, H. A.|textbf
Pizzo, R. P.
observatories!ground-based
archives!individual!LOFAR
data!management!operations

Sustainable data life cycle management for data intensive instruments

Hanno Holties and Roberto Pizzo

NWO-I ASTRON, Dwingeloo, the Netherlands; holties@astron.nl

Abstract. LOFAR is a data intensive facility that faces severe challenges in data processing and storage. Since start of science operations, LOFAR has accumulated over 50 petabytes of data in its science data archive and following a major upgrade of the instrument, it is expected that over the course of the next five years of operations the archive will grow to well over 100 petabytes of science data. Based on experiences from the first phase operations, a spectrum of measures is presented that address the sustainability challenges for LOFAR.

1. Sustainability challenges

The throughput of digitized antenna signals for the LOFAR instrument (van Haarlem et al. 2013) amounts to two terabytes per second, which is reduced in several stages by a factor of circa one million for a fully processed image survey data release. However, the current data archive consists mainly of partly reduced data products that are used by science teams for further processing and analysis. Since the start of science operations in 2012, LOFAR has accumulated over 50 petabytes of data in its science data archive. The instrument currently is undergoing a major upgrade, following which it is expected that over the course of the next five years of operations the archive will grow to well over 100 petabytes of science data, which would include ca 50 petabytes of advanced data products. Other astronomical instruments that will start and ramp up over the coming years will generate similar (RUBIN) or significantly larger (SKA) data volumes. Although there are viable technical solutions to scale storage to the required capacity, sustainability considerations (carbon footprint, storage cost, network and processing capacity, data access and re-use by a wide community) are increasingly putting constraints on acceptable volumes for data delivery and long-term storage. In the following sections a spectrum of measures is presented that are applied to the data systems and operations to address the sustainability challenges for LOFAR. NB In this paper, the instrument before the upgrade is referred to as 'LOFAR1' while the instrument after the upgrade is referred to as 'LOFAR2.0'.

2. Lessons from LOFAR1

During the first five years of operations (2012-2017), the LOFAR observatory and the community needed to get to grips with the huge data volumes generated by a revolutionary new type of instrument. Up to seven petabytes of data were added to the data archive each year. Of these, initially only a fraction was accessed while appropriate

calibration and data reduction algorithms were developed. After about three years, the processing pipelines reached a maturity level that allowed to process instrument data from the archive at roughly the same rate as it was generated. However, over the first years of operations various improvements for the instrument and it's data processing capabilities were implemented, leaving a significant fraction of the archival data from the early phase to be considered of lower quality, limiting the legacy value of early cycle data.

The next phase of LOFAR1 operations (2018-2024) saw the generation of a steady output of good quality data and growing impact of the instrument, where both publications and community increased with a factor of more than three while maintaining a stable observing efficiency of circa 70%. From 2018, compression algorithms were applied to archived visibility data, which resulted in more than halving the total volume of archived data. However, the LOFAR1 archive was still primarily used to keep intermediate level data products which saw only limited re-use throughout the lifetime of LOFAR1.

The Carbon footprint and energy consumption of LOFAR1 was assessed in 2023 (Kruithof et al. 2023). For science and data processing, it was found that more than 40% of the energy consumed over the data lifecycle was spent on data processing. The most demanding type of data processing is widefield long baseline imaging, which is of critical importance for key science cases. However, the required level of compute resources severely limited the data that could be imaged fully. And although the energy footprint of archival storage is only 2% of the total, storing large data volumes for long term does result in significant overall operational cost for the instrument.

3. Addressing the challenges for the LOFAR data life cycle

To address the sustainability challenges, several actions are being taken. First, the footprint of the LOFAR1 data is being reduced through a multi-pronged approach under the flag of the LOFAR Data Valorisation (LDV) project (Bozkurt et al. 2024). As legacy data is touched, value is being added to the archive by correcting known data issues and extracting metrics from the data. Within LDV, the following actions are taken:

- Retire archived data from early LOFAR cycles that do not offer legacy value.
- Apply a scalable data processing framework for editing archive content at tens of petabytes scale (Holties et al.).
- Extract quality information from archived data and add this to the archive catalog.
- Apply compression to data from before the availability of the compression techniques.
- Reduce the number of quantization bits for time-series data (pulsar research).
- Re-balance the data storage across the three data centers that contribute storage capacity to the LOFAR Long-Term Archive.

Before enforcing retirement of early cycle data, the LOFAR community has been consulted to ascertain which data represents legacy value. Teams were allowed to retrieve data they would prefer to keep a copy of and others applied for retaining intermediate

level data for legacy purposes. Following an appraisal of the requests, an agreement was reached with selected teams to retain data, but reduce the volume by applying appropriate methods. By the end of 2025, the LDV project has realized a reduction of the total volume of the LOFAR1 archive by more than 15 petabytes.

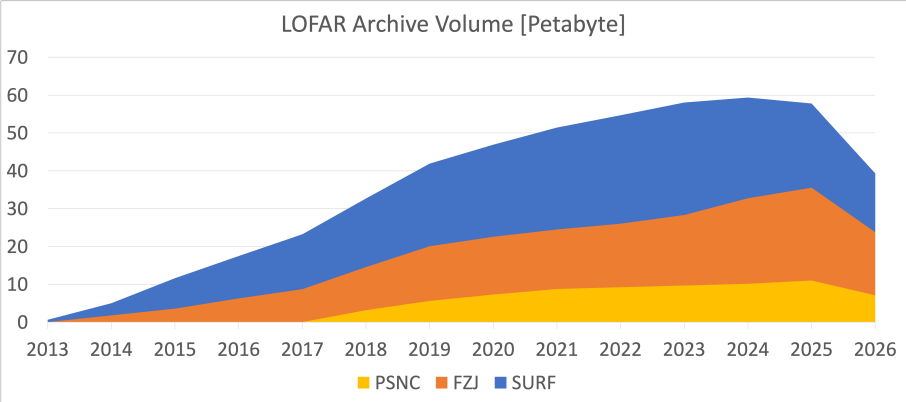


Figure 1. The volume of the LOFAR1 archive, including distribution over the three sites: SURF (NL), ForschungsZentrum Juelich (FZJ, DE), Poznan Super Computer Centre (PSNC, PL).

At the same time, it was realized that for LOFAR2.0 a shift of paradigm was needed and science operations would primarily target the generation and archiving of advanced data products, as opposed to indefinitely keeping lower level data products. This resulted in the following initiatives:

- Plan for the generation of advanced data through pipelines, built together with the community, with the longer term goal to execute these under central control.
- Implement a policy that targets retirement of intermediate level data products after a finite time, while advanced data products are to be returned to the archive and retained for the longer term.
- Work with the community to define the standard data products that are to be generated and archived.
- Ask all science teams to prepare data management plans that detail the data lifecycle from observation up to release of curated data collections with an associated publication plan.

The data product definitions and data management plans allow the observatory to effectively prepare and operate the data archive for LOFAR2.0. At the same time, data operations are developed to improve and automate data curation processes, quality assurance, and further FAIRification of the data archive.

4. Advances in data processing for LOFAR

An important contribution to realizing sustainabilty of LOFAR data operations is coming from advances in the data processing software that is used to reduce LOFAR data. A

number of data compression techniques have now become available for LOFAR image processing workflows:

- Dysco (lossy) compression of visibilities (Offringa 2016), resulting in a volume reduction with circa a factor four with negligible effect on science content (larger factors may be achievable, depending on science case).
- Lossless compression of embedded metadata tables, resulting in a typical further volume reduction of 12% for visibility data products.
- Sisco (lossless) compression of intermediate model data (Offringa & van Weeren 2026), resulting in a typical size reduction for model data with a factor four.

For timeseries data, lossy rebitting from 8 to 2 bits has been demonstrated to be possible while maintaining use of the data for pulsar search and characterization. Other advanced techniques have been developed to reduce long-term storage and at the same time significantly reduce computational costs for wide field long-baseline image processing (de Jong et al. 2025):

- Baseline-Dependent Averaging where data for shorter baselines is stored at lower resolution.
- Sidereal Visibility Averaging for fields observed multiple times, applying averaging of data for similar baselines is applied before processing.

Together, the new techniques allow reducing compute costs with a factor four on average for a typical eight hour observation. Further potential gains have been identified.

Acknowledgments. The authors would like to thank all members from the LOFAR community that have contributed to the solutions developed and applied over time for addressing LOFAR data challenges. They would like to thank Marco Iacobelli, Vlad Kondratiev, André Offringa, John Swinbank and Stefan Wijnholds for sharing material and feedback.

References

Bozkurt, A., et al. 2024, in Observatory Operations: Strategies, Processes, and Systems X, edited by C. R. Benn, A. Chrysostomou, & L. J. Storrie-Lombardi, International Society for Optics and Photonics (SPIE), vol. 13098, 130982P. URL https://doi.org/10.1117/12.3019088

de Jong, J. M. G. H. J., et al. 2025, Monthly Notices of the Royal Astronomical Society, 542, 3253. URL https://doi.org/10.1093/mnras/staf1373

Holties, H. A., et al.in ADASS XXXIV, edited by K. Polsterer, & S. Wagner (San Francisco: ASP), vol. TBD of ASP Conf. Ser., 999 TBD

Kruithof, G., et al. 2023, Exp Astron, 56, 687–714. URL https://doi.org/10.1051/0004-6361/201629565

Offringa, A. R. 2016, A&A, 595, A99. URL https://doi.org/10.1051/0004-6361/201629565

Offringa, A. R., & van Weeren, R. J. 2026, A&A, 706, A68. URL https://doi.org/10.1051/0004-6361/202557822

van Haarlem, M. P., et al. 2013, A&A, 556, A2. eprint: 1305.3550

Astronomical Data Analysis Software and Systems XXXV
*ASP Conference Series, Vol. **VOL** Compiled: Tuesday 30th June, 2026@ 01:30*
K. Polsterer, D. Bohlender, S. Gwyn, A. Hincks and P. Teuben, eds.
*© **YEAR** Astronomical Society of the Pacific*

ZZZDEBUG!107.tex
Kirsch, C.|textbf
Lorenz, M.
Dauner, L.
Gülbahar, E. G.
Reinmann, N.
Pal, K.
Stierhof, J. J. R.
Thalhammer, P.
Dauser, T.
Wilms, J.
software!simulation
astronomy!X-ray
computer languages!C++
software!development
software!design
SIXTE! ascl:1903.002

Refactoring the SIXTE simulator: Towards a more modular code base

C. Kirsch, M. Lorenz, L. Dauner, E. G. Gülbahar, K. Pal, N. Reinmann, J. J. R. Stierhof, P. Thalhammer, T. Dauser and J. Wilms

Dr. Remeis-Sternwarte and ECAP, Friedrich-Alexander-Universität Erlangen-Nürnberg, Sternwartstr. 7, Bamberg, Bavaria, Germany; christian.ck.kirsch@fau.de

Abstract. The SIXTE (SImulation of X-ray TElescopes) software is a general end-to-end simulation toolkit for X-ray observations, covering the full observation process from source photon generation to detector readout and the production of high-level output files. It is the official simulator for existing and future X-ray missions, such as *eROSITA*, *NewAthena*, *THESEUS* and *AXIS*.

Originally being designed as a simulator for *eROSITA*, the addition of new instrument and telescope types over several years has made the original code base increasingly difficult to maintain. As such, we have refactored the code, changing languages from C to C++ and switching to a more modular software design to facilitate the implementation of new models. This proceeding highlights some of the design choices used during the refactoring as well as its effects on maintenance and new feature development one year after release of the refactored code base.

1. Introduction

The SIXTE (Dauser et al. 2019) software is an open source[1] simulation toolkit for X-ray observations, with the goal of accurately simulating both existing and future X-ray missions in order to assess their scientific performance. To achieve this, SIXTE generates source photons based on a flexible source definition format called SIMPUT. Photons are then imaged onto the focal plane of a telescope using either raytracing or stored images of the telescope point spread function (PSF). This imaging also takes into account effects due to off-axis pointings, e.g., during surveys, such as vignetting and PSF degradation. The instrument models of SIXTE also take into account non-linear detection effects such as pile-up. Figure 1 shows an example simulation with these effects.

First versions of SIXTE were developed in the mid 2000's as separate tools for simulating X-ray observations. Over the following roughly 20 years, the scope of the software was extended, starting with the simulation of CCD instruments like *eROSITA* (Predehl et al. 2021), and going to DEPFET detectors like the *NewAthena Wide Field Imager (WFI)* (Meidinger et al. 2021), microcalorimeters like the *NewAthena X-ray Integral Field Unit (X-IFU)* (Peille et al. 2025) and other detector types.

[1] https://www.sternwarte.uni-erlangen.de/sixte/

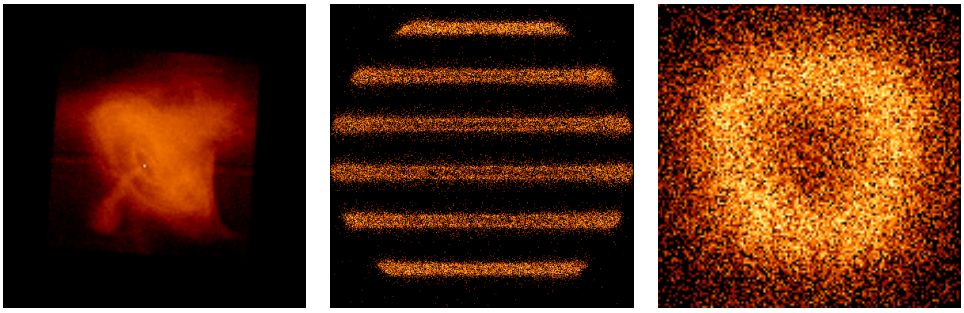


Figure 1. Example simulation showing the capabilities of SIXTE for an extremely bright source. *Left:* Input photons generated from a model containing the Crab Nebula and Pulsar. *Middle:* Photons detected by one CCD of *eROSITA* observing in survey mode, in raw detector coordinates. *Right:* Detected photons reprojected into sky coordinates. The overall image is distorted by the optics of *eROSITA*, with a loss of counts towards the center of the image due to pile-up.

The majority of these new detector types and other features was implemented within the scope of academic theses (e.g., Schmid (2012)) by more than twenty contributors, who were mostly astrophysicists. Both the regular fluctuation of the developer base and development constraints by thesis and instrument deadlines led to several design decisions that, while effective in the short term, hampered the long-term development of the code base, especially with regards to adding new features.

With the intended addition of new missions such as the *Advanced X-ray Imaging Satellite (AXIS)* (Reynolds et al. 2023) and the *Transient High-Energy Sky and Early Universe Surveyor (THESEUS)* (Amati et al. 2018), we decided that the SIXTE code base needed to be refactored to be more extensible and maintainable. The rest of this proceeding will discuss the process of this refactoring and its outcomes.

2. Refactoring

2.1. Issues with the code base

One of the primary issues of the old SIXTE code base was code duplication and inefficient re-use of existing functionality. A primary example of this was the existence of individual simulation tools for several instruments. Aside from a general simulator for CCD-based instruments, called `runsixt`, additional simulators existed for *eROSITA* (composed of seven individual telescopes with one CCD focal plane each), *NewAthena WFI* (using a single telescope with a focal plane composed of four DEPFET detectors) and *NewAthena X-IFU* (using a single telescope with a microcalorimeter detector). While these tools differed in details such as the number of detectors or types of detectors simulated, they shared many other tasks, such as the generation of input photons, or reprojection of photons from detector to sky coordinates.

Some of these tasks, such as photon generation, were separated into their own functions and thus reusable, but many others, such as photon reprojection were not, due to slight differences in, e.g., detector geometries. In addition, when implementing new simulation tools, much of the code used to initialize structures like those used for photon

generation was copied from existing tools to new tools (with slight modifications, if necessary), including memory allocation and error handling.

In practice, this separation of the simulator into different tools made it difficult to add new features, as it often required editing multiple simulation tool chains with sometimes subtle differences that an individual developer was not aware of. One of the main goals of the refactoring was the elimination of cases like these.

2.2. Refactoring goals and process

There were five main goals for the refactoring process:

1. Code duplication – examples of which were shown above – should be reduced, to make the code more maintainable.

2. The overall design of the code should be opened up for extensions. This was mainly motivated by future missions with new detector types that would need to be supported.

3. The output files of the code should stay compatible with previous versions. Users of SIXTE often process its output with custom-written scripts, which should not be affected by this refactoring.

4. There should be no regressions in performance, particularly in run time.

5. As SIXTE is the official simulator of several future X-ray missions, it must still be possible to compile and run the code during the lifetime of these missions, which may be more than 20 years into the future.

The first decision made in the refactoring process was to port the code base from C to C++. This choice essentially helps to fulfill all of the mentioned goals: Code duplication can be reduced in C++ by proper use of the `std` library, as well as using constructors and destructors for memory management. Correct use of templates and `virtual` interfaces further reduces code duplication and also serve to make a code base more extensible. In terms of performance, C and C++ are generally equivalent, and more so dominated by algorithmic complexity. Being an ISO standardized language, a C++ project can also be expected to still compile during the lifetime of the missions SIXTE supports.

To ensure compatibility with the output of previous versions, we created a comprehensive testing setup. These tests checked both the direct output event files of SIXTE as well as derived products used by scientists, such as spectra and images. The tests were automatically run via GitLab continuous integration on every git commit, with merges to the main development branch only allowed when all tests passed. While not done automatically, these tests could also be used to discover possible performance regressions.

To properly make use of C++, all developers attended dedicated workshops on C++ software design. Following these workshops, we then first created an outline object hierarchy which would encompass the main tasks of the old SIXTE version, such as photon generation, photon imaging and photon detection. These objects are then used by a new tool called `sixtesim`, which replaces the individual simulator tools mentioned in Sec. 2.1. To achieve this, `sixtesim` can flexibly simulate multiple telescopes at once

(as needed for *eROSITA*), which can each individually focus photons onto multiple detectors (as needed by *NewAthena WFI*) of different types, such as CCD or DEPFET detectors and microcalorimeter arrays. This was made possible by unifying the previously separate objects for simulating these different detector types behind an abstract `Detector` object using shared interfaces, but a separate internal implementation.

As a test of this hierarchy, we then implemented a full instrument setup – in our case, a CCD detector based on *eROSITA*. This implementation helped diagnose any possible issues with the object hierarchy before adding more detectors to it.

Once this first simulation chain was implemented, we could build out the object hierarchy by adding more detector types and develop further tools using this object hierarchy. As the interfaces to all objects were defined ahead of time, this could be done in parallel by multiple developers.

3. Summary and conclusions

The implementation of the refactored SIXTE version took approximately two years, finishing in November 2024 with the public release of SIXTE version 3.0.3. Development has continued since, with the latest version at the time of writing being version 3.3.2 in January 2026.

Aside from bug fixes, several new features were added in this time, mostly related to the *THESEUS* mission, which uses both optics (lobster eye, coded mask) and detector types (silicon drift detectors) previously not supported by SIXTE. For the optics, a new raytracing module was implemented, which can be used interchangeably with the old imaging implementation, based on the sampling of PSFs.

In terms of performance, the stated goal of preventing regressions was actually exceeded – a review of the old code showed several inefficient algorithms and unnecessary memory allocations. Improving these aspects of the code led to an improvement of the simulation speed by a factor of two to ten, depending on the used instrument and simulation case. Testing during the refactoring also revealed several undiscovered bugs in the old SIXTE code, which were not reproduced in the new implementation.

In conclusion, the refactoring of the SIXTE code can be considered a success, providing a new code base that is much easier to maintain and extend, making it more robust for long term development and the inclusion of new X-ray instruments.

References

Amati, L., et al. 2018, Advances in Space Research, 62, 191. eprint: 1710.04638

Dauser, T., et al. 2019, A&A, 630, A66. eprint: 1908.00781

Meidinger, N., et al. 2021, in Society of Photo-Optical Instrumentation Engineers (SPIE) Conference Series, edited by J.-W. A. den Herder, S. Nikzad, & K. Nakazawa, vol. 11444 of Proc. SPIE, 114440T

Peille, P., et al. 2025, Experimental Astronomy, 59, 18. eprint: 2502.10866

Predehl, P., et al. 2021, A&A, 647, A1. eprint: 2010.03477

Reynolds, C. S., et al. 2023, in UV, X-Ray, and Gamma-Ray Space Instrumentation for Astronomy XXIII, edited by O. H. Siegmund, & K. Hoadley, vol. 12678 of Proc. SPIE, 126781E. eprint: 2311.00780

Schmid, C. 2012, Ph.D. thesis, Friedrich-Alexander-Universität Erlangen-Nürnberg

Astronomical Data Analysis Software and Systems XXXV
*ASP Conference Series, Vol. **VOL** Compiled: Tuesday 30th June, 2026@ 01:30*
K. Polsterer, D. Bohlender, S. Gwyn, A. Hincks and P. Teuben, eds.
*© **YEAR** Astronomical Society of the Pacific*

ZZZDEBUG!109.tex
Ates, O.|textbf
Ates, O.
Servillat, M.
Boisson, C.
Cristofari, P.
Lamiré, B.
Savalle, R.

Accessing and Visualizing Cherenkov Data via an Open Science Web Platform

O. Ates[1], M. Servillat[1], C. Boisson[1], P. Cristofari[1], B. Lamiré[2], R. Savalle[2]

[1] *LUX, Observatoire de Paris, Université PSL, Sorbonne Université, CNRS, 92190 Meudon, France;* onur.ates@obspm.fr

[2] *DIO, Observatoire de Paris, 92190 Meudon, France*

Abstract. We are developing an open-source web platform that enables data search, visualization and preview jobs of Cherenkov Astronomy data products. This platform manages all the necessary metadata to support search criteria adapted to the specific nature of Cherenkov data. Built on the Virtual Observatory framework, the platform implements the IVOA ObsCore standard to describe the spatial, temporal, and spectral characteristics of observations. Additionally, we are contributing to a High-Energy extension of ObsCore, enhancing its capacity to describe high-energy Cherenkov data.

1. Cherenkov Data and IVOA Standards

Arrays of ground based imaging Cherenkov telescopes (IACT), as H.E.S.S., MAGIC and VERITAS, detect ultra-short (nanosecond) flashes of blue light produced when high-energy gamma-rays hit Earth's atmosphere, triggering particle cascades. The upcoming Cherenkov Telescope Array Observatory (CTAO) will generate hundreds of petabytes of raw data annually, requiring extensive atmospheric monitoring and rich metadata to reconstruct science-ready data products such as event lists, images, spectra, and light curves. These software telescopes depend on complex pipelines and statistical models, not just the instrument properties and acquisition settings. The full processing configuration and detailed provenance is thus critical information for the end user to enable the scientific analysis of such data.

Open Science is now commonly illustrated by the FAIR principles for data management: Findability, Accessibility, Interoperability, and Reusability. The integration of International Virtual Observatory Alliance (IVOA) standards into Cherenkov astronomy will help to create searchable public data archives that follow the FAIR principles (Servillat et al. 2022). These archives support both structured metadata and rich datasets, making it easier for users to find and access the data they need.

We adopt the IVOA Observation Data Model Core Components (ObsCore, Louys et al. 2017) to index spatial, temporal, and spectral metadata, ensuring that Cherenkov data can be consistently described, stored, and shared across different observatories and data archives. This fosters interoperability between datasets from various instruments and observatories, facilitating broader collaboration within the astronomical community. The IVOA Table Access Protocol (TAP) is used to allow users to query large datasets remotely using the Astronomical Data Query Language (ADQL). This protocol is especially useful for Cherenkov astronomy, where datasets can be large and

complex. With TAP, researchers can efficiently query datasets for specific criteria (e.g., time, energy range, sky location) and obtain relevant results.

The platform aligns also with the High-Energy (HE) ObsCore extension under development within the IVOA High Energy Interest Group (HEIG), adding fields tailored to High Energy data (see the dedicated IVOA Note[1]). By then expressing Cherenkov-specific metadata (e.g., energy bounds, zenith angle, muon efficiency, safe analysis range) in a common HE-ObsCore schema, the platform will issue the same TAP/ADQL queries across future CTAO and current IACT archives, enabling uniform discovery and precise filtering, regardless of the original archive or file naming.

We already implemented VO services to access H.E.S.S. public test data release[2], though without a dedicated web interface. A survey of various data access platforms has been conducted to propose an initial design (e.g. ALMA Science Archive[3]).

2. Implementation of Cherenkov data access

This contribution presents an open science platform[4,5] to discover and visualize IACT data, and get some insight on a selection of data products via preview jobs.

Web Platform Architecture. We use FastAPI for the backend, with endpoints that forward ADQL queries to an external TAP service, object-resolve requests to SIMBAD/NED catalogues[6,7], and (for signed-in users) Basket, Query Store, as well as Preview Job submissions to the a job management service (using the Observatoire de Paris UWS System, OPUS, Servillat et al. 2022). Redis caches ADQL results and object lookups and holds session data to reduce repeated external calls and resource usage. The frontend is built with React. For visualization we embed Aladin Lite (interactive sky map with selectable markers) and Plotly (time coverage and energy-range plots). Styling uses Bootstrap.

Search and Visualization. Snapshots are shown in Figure 1.

In the current version, the user has the option to query with coordinates and/or observation dates via the search form. The time search area has the option to choose between time systems. Filling one field automatically fills the other ones. In order to answer use cases specific to Cherenkov astronomy, more criteria will be integrated to the search form, e.g.: data level, proposal ID, science target ID, object type, average azimuth, elevation, Moon level, live time, telescope/array configuration, pointing mode.

Search results are rendered in a sortable table with default columns on display (the remaining columns can be selected to show via Toggle Columns menu). The results are synchronized with an Aladin Lite[8] view where each row appears as a marker with

[1] https://www.ivoa.net/documents/Notes/VOHE/
[2] https://hess-dr.obspm.fr/
[3] https://almascience.eso.org/aq/
[4] https://padc-ctao-data-explorer.obspm.fr
[5] https://gitlab.obspm.fr/oates/ctao-data-explorer
[6] https://simbad.cds.unistra.fr/
[7] https://ned.ipac.caltech.edu/
[8] https://aladin.cds.unistra.fr/AladinLite/

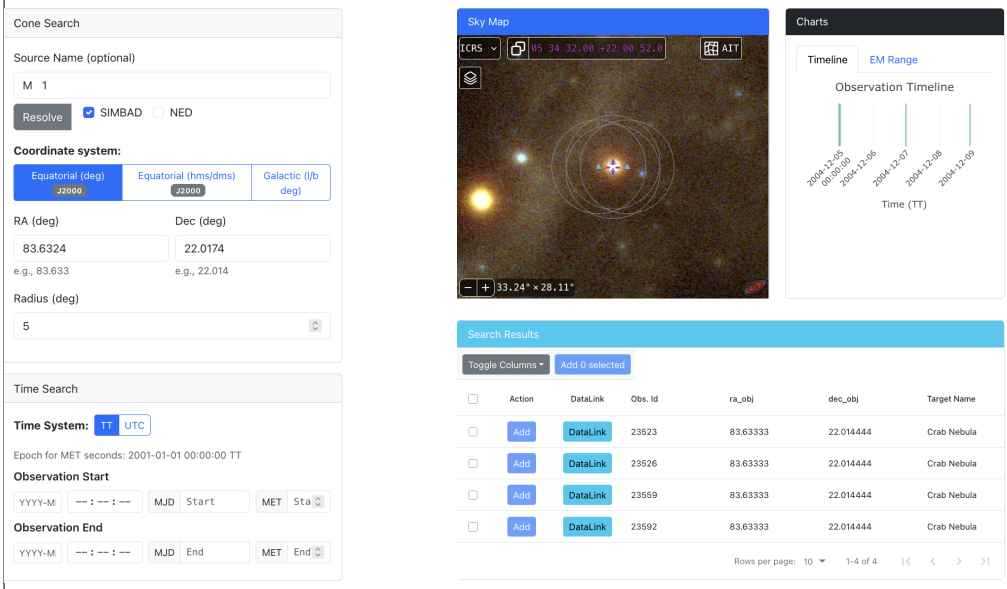


Figure 1. Data Explorer snapshots. Left: search form with cone search, object-resolve, and time filters; Right: results page with synchronized table, sky map, and timeline chart.

a field-of-view (FoV) circle. Results also appear on Plotly-based timeline and energy-range charts. Colors use a color-blind–safe palette.

User Features. Authentication is based on OpenID Connect and federated identity. We store minimal user information in a PostgreSQL database (a user identifier); we do not persist names or email addresses. We also support reproducible workflows: every ADQL query is stored and can be reloaded or deleted via the Query Store. Users can select items in the results table, sky map, or charts. Selected records can be added to persistent Baskets for later reuse, visualization, or Preview Jobs. Anonymous users retain session-scoped state. Those features ensure a better traceability of actions and efficiency of the platform.

Preview Jobs. Signed-in users can submit a Basket group to a job management service. An active Basket shows a Run Preview Job button that opens a form with job parameters. On submission, the observation IDs in the Basket are passed to a Gammapy[9] source-analysis script running asynchronously on the job server. Users are redirected to a job-details page showing job phase (queued, executing, completed, or error) and logs. On success, the results include figures such as stacked excess maps and stacked source spectra, enabling rapid assessment of the selected data.

Testing and Monitoring. We exercise the backend with `pytest` at unit, integration, and end-to-end levels. External services (TAP, SIMBAD/NED, OPUS/UWS) are stubbed where appropriate for determinism; session flows (cookie/Redis, token refresh,

[9] `https://gammapy.org/`

logout), `/users/me` profile hydration, query-history persistence, Basket operations, and Preview-Job submission paths are covered. On each push, *GitLab CI* runs the test suite and triggers `SonarQube` analysis. Formatting with `Black`, linting with `Ruff`, and static typing with `mypy` are run locally using the project's `pyproject.toml` configuration before pushing changes.

The service is deployed to *staging* and *production* behind `nginx` (reverse proxy) with `gunicorn` workers serving the FastAPI application. Both environments expose Prometheus-compatible metrics; *Grafana* dashboards track availability, request rates, p95 latencies for VO/TAP/SIMBAD calls, OPUS job submissions/completions, cache hit/miss, and Redis usage. This setup lets us trial changes on staging while continuously observing their impact before promoting to production.

3. Roadmap

Upcoming work includes (i) frontend tests; (ii) additional search criteria (e.g., data product type, data level, proposal ID, average azimuth/elevation); (iii) deployment of DataLink 1.1 services; (iv) enhancements to Preview Jobs (e.g., background map in Aladin Lite or JS9 with interactive contrast/scale); (v) containerized deployment with Docker—Dockerfiles (backend/front end) are already in the repository and tested locally; next steps are CI builds of versioned images and a compose-based stack on staging before promoting to production; and (vi) integration with the CTAO portal and related services.

Acknowledgments

We thank the CTAO community, colleagues contributing to the IVOA High Energy Interest Group and the support from the Paris Astronomical Data Center (PADC).

CTAO: `https://www.ctao.org/`
H.E.S.S.: `https://www.hess-experiment.eu/`
MAGIC: `https://magic.mpp.mpg.de/`
VERITAS: `https://veritas.sao.arizona.edu/`

References

Louys, M., Tody, D., Dowler, P., Durand, D., Michel, L., Bonnarel, F., Micol, A., & IVOA DataModel Working Group 2017, Observation Data Model Core Components, its Implementation in the Table Access Protocol Version 1.1, IVOA Recommendation 09 May 2017

Servillat, M., Aicardi, S., Cecconi, B., & Mancini, M. 2022, in Astronomical Society of the Pacific Conference Series, edited by J. E. Ruiz, F. Pierfedereci, & P. Teuben, vol. 532 of Astronomical Society of the Pacific Conference Series, 451. eprint: arXiv:2101.08683

Servillat, M., Boisson, C., Fuessling, M., & Khelifi, B. 2022, in Proceedings of the 31st Annual Conference on Astronomical Data Analysis Software and Systems (ADASS XXXI), vol. XXXI of ASP Conference Series. Also available on arXiv: `https://arxiv.org/abs/2201.03247`, URL `https://www.aspbooks.org/a/volumes/article_details/?paper_id=40857`

Astronomical Data Analysis Software and Systems XXXV
*ASP Conference Series, Vol. **VOL** Compiled: Tuesday 30th June, 2026@ 01:30*
K. Polsterer, D. Bohlender, S. Gwyn, A. Hincks and P. Teuben, eds.
*© **YEAR** Astronomical Society of the Pacific*

ZZZDEBUG!117.tex
Schussler, F.|textbf
Ashkar, H.
KiendrÃ©bÃ©ogo, W.
Seglar-Arroyo, M.
de Bony, M.
Ruiz-Velasco, A. B. E.
Le Montagner, R.

`tilepy`: A Flexible Open-Source Scheduling Engine for Time-Domain and Multi-Messenger Astronomy

Fabian Schussler,[1] H. Ashkar,[2] W. Kiendrébéogo,[1] M. Seglar-Arroyo,[3] M. de Bony[1,4], A. Berti[5] E. Ruiz-Velasco[6], and R. Le Montagner[7]

[1]*IRFU, CEA, Université Paris-Saclay, France;* `fabian.schussler@cea.fr`
[2]*ICE-CSIC, Barcelona, Spain* [3]*IFAE, Barcelona, Spain* [4]*CPPM, Marseille, France* [5]*MPP, Munich, Germany* [6]*LAPP, Annecy, France* [7]*IJCLab, Orsay, France*

Abstract. The era of multi-messenger astrophysics requires rapid and efficient follow-up of transient events, many of which, such as gravitational waves (GW), gamma-ray bursts (GRB), and high-energy neutrinos, suffer from poor sky localisation. We present `tilepy`, a Python-based software designed to optimize observation schedules for these events. We here detail the modular architecture of `tilepy`, which separates high-level scheduling logic from low-level tiling and pointing tools, enabling full adaptability for ground- and space-based observatories. Furthermore, we describe the integration of `tilepy` into the Astro-COLIBRI platform, providing the community with a user-friendly interface and API for triggering complex observation campaigns in real time.

1. Introduction

Time-domain astrophysics has been transformed by the direct detection of gravitational waves and high-energy neutrinos. However, a significant challenge remains: the localization uncertainty of these transients often spans tens to thousands of square degrees (Abbott et al. 2023; Abac et al. 2025). Since most electromagnetic follow-up facilities possess fields of view (FoV) significantly smaller than these error boxes, efficient tiling strategies are essential to maximize the probability of detecting a counterpart.

`tilepy` was developed to address this optimization problem. It maximizes the probability of counterpart detection by leveraging both, 2D skymaps as well as 3D galaxy catalogs, while strictly adhering to observatory constraints such as visibility, background light conditions, and instrument limits (Seglar-Arroyo et al. 2024). It is available under the GNU Lesser General Public License via GitHub (Schüssler et al.).

While previous works have demonstrated `tilepy`'s application to specific campaigns like GW170817 (Abdalla et al. 2017) and other GW follow-up observations (Abdalla et al. 2021), we here focus on the software's internal architecture and its integration into the broader multi-messenger ecosystem via Astro-COLIBRI.

2. Software Architecture

The design of `tilepy` follows a modular, hierarchical structure. This ensures maintainability and allows users to customize strategies without altering the core scheduling

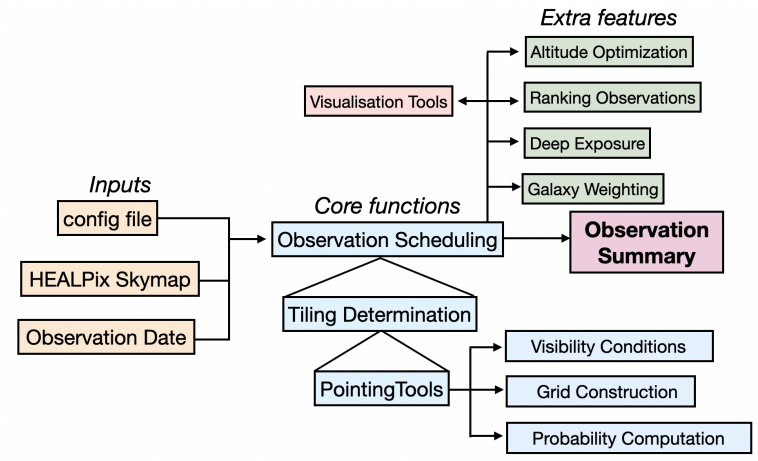


Figure 1. Outline of the `tilepy` architecture. From Seglar-Arroyo et al. (2024)

logic. The architecture is illustrated in Figure 1 and can be categorized into three primary layers: Inputs, Core Functions, and Outputs.

2.1. Inputs and Configuration

The software requires three primary inputs to initialize a campaign. First, a localization probability map of the transient (e.g., from LIGO/Virgo/KAGRA, Fermi-GBM, or IceCube). Typically given in *HEALPIX* format. Second, the proposed start time for the campaign and finally, a distinct configuration file is required for each participating telescope. These files define the observatory's location, FoV size and shape, horizon limits, and specific constraints (e.g., Moon avoidance angles). This modularity allows `tilepy` to schedule multi-observatory campaigns simultaneously by simply loading multiple configuration files.

2.2. Core Functions

The processing logic is divided into three levels of abstraction, moving from global, high-level scheduling to low-level computation:

2.2.1. Observation Scheduler

This is the top-level manager. It parses the configuration files and the input skymap. It determines the appropriate strategy (e.g., 2D probability integration vs. 3D galaxy targeting) based on the available data (e.g., whether the event includes distance information) and user requests. By default, the scheduler employs a "greedy" approach: it identifies the highest-probability region available at the earliest possible time slot, schedules it, masks that region to prevent duplicate observations, and iterates until the observing window closes or probability thresholds are met. Details are given in (Ashkar et al. 2021).

2.2.2. Tiling Determination

The tiling layer handles the geometry of the search. It supports varying grid resolutions, balancing computational speed with accuracy. Users can specify the required resolu-

tion (via the HEALPix `NSIDE` parameter) or allow the software to optimize it based on the telescope's FoV. Recent updates have expanded this layer to support complex FoV shapes. Beyond standard circular fields, `tilepy` now supports polygonal shapes (squares, hexagons) and arbitrary rotations, which is critical for accurate modeling of instrument footprints like those of the Vera Rubin Observatory, Swift-XRT, etc.

2.2.3. Pointing Tools

This low-level layer performs the mathematical heavy lifting. Its core components include *Visibility Handling*, which computes the altitude/azimuth of targets over time, checking against constraints (Sun/Moon altitude, Earth limb for satellites). The *Probability Computation* integrates probability density over the defined FoV. For 3D strategies, it queries galaxy catalogs (e.g., GLADE+ (Dálya et al. 2022)) to weigh pointings by stellar mass or galaxy luminosity. New modules supporting *Space-Based Observatories* have been added recently to calculate spacecraft trajectories, identifying occultations by the Earth, Sun, and Moon, and flagging passages through the South Atlantic Anomaly (SAA).

2.3. Outputs and Visualization

The final layer generates the observation schedule (coordinates and timing) and a suite of visualization tools. These include coverage maps, visibility plots (altitude vs. time), and probability accumulation curves, aiding observers in assessing the quality of the schedule.

3. API and integration with Astro-COLIBRI

To make advanced scheduling accessible to the wider community, `tilepy` has been integrated into the Astro-COLIBRI platform (Reichherzer et al. 2021, 2023). This integration bridges the gap between alert reception and observation planning.

The core functionality of `tilepy` is hosted as a public API [1]. This decoupling allows external services to request schedules without installing the software or its dependencies locally. The API accepts an event identifier (or a skymap URL) and observatory parameters, and returns a JSON-formatted schedule. This architecture supports the "Trigger" model, in which automatic transient handlers can programmatically request observation plans immediately upon receiving an alert.

For manual users, `tilepy` is embedded directly into the Astro-COLIBRI web and mobile interfaces. Users viewing an event in Astro-COLIBRI can generate an optimized tiling plan with a single click. This integration democratizes access to complex scheduling algorithms and allows citizen scientists to visualize the visibility of a GW event and obtain a mathematically optimized pointing list without writing code. The platform also hosts a dedicated helpdesk and discussion forum[2] to support `tilepy` users.

[1] `https://tilepy.com`
[2] `https://forum.astro-colibri.science/c/instrumentation-and-tools/tilepy`

4. Status and Future Outlook

The `tilepy` ecosystem is fully operational and utilized by major collaborations like H.E.S.S., and the first CTAO Large-Sized Telescope (LST-1). The connection to Astro-COLIBRI and the public API ensure that the optimized observation schedule are widely available and easily accessible also by citizen scientists and amateur astronomers.

`tilepy` provides a robust, modular solution for the complex problem of multi-messenger follow-up. Its architecture, separating configuration from core logic, allows it to serve a diverse range of facilities, from ground-based Cherenkov telescopes to space-based X-ray observatories. The integration with Astro-COLIBRI represents a significant step forward in usability, allowing real-time access to high-performance scheduling tools. Future developments will focus on further optimizing algorithms for "Travelling Salesman" style slew-minimization (possibly using AI aided approaches), expanding support for non-HEALPix localization formats.

Acknowledgments

We acknowledges the support of the French Agence Nationale de la Recherche (ANR) via the project "Multi-messenger Observations of the Transient Sky (MOTS)" under reference ANR-22-CE31-0012. We also acknowledge support from the "Astrophysics Centre for Multi-messenger studies in Europe (ACME)" funded by the European Union's Horizon Europe Research and Innovation programme under grant no. 101131928.

References

Abac, A. G., Abouelfettouh, I., Acernese, F., Ackley, K., Adhicary, S., Adhikari, D., Adhikari, N., Adhikari, R. X., Adkins, V. K., Afroz, S., & et al. 2025, ApJ, 995, L18. eprint: 2508.18080

Abbott, R., Abbott, T. D., Acernese, F., Ackley, K., Adams, C., Adhikari, N., Adhikari, R. X., Adya, V. B., Affeldt, C., Agarwal, D., & et al. 2023, Physical Review X, 13, 041039. eprint: 2111.03606

Abdalla, H., Abramowski, A., Aharonian, F., Ait Benkhali, F., Angüner, E. O., Arakawa, M., Arrieta, M., Aubert, P., Backes, M., Balzer, A., & et al. 2017, ApJ, 850, L22. eprint: 1710.05862

Abdalla, H., Aharonian, F., Ait Benkhali, F., Angüner, E. O., Ashkar, H., Backes, M., Baghmanyan, V., Barbosa Martins, V., Batzofin, R., Becherini, Y., & et al. 2021, ApJ, 923, 109. eprint: 2112.08307

Ashkar, H., Brun, F., Füßling, M., Hoischen, C., Ohm, S., Prokoph, H., Reichherzer, P., Schüssler, F., & Seglar-Arroyo, M. 2021, JCAP, 2021, 045. eprint: 2010.16172

Dálya, G., Díaz, R., Bouchet, F. R., Frei, Z., Jasche, J., Lavaux, G., Macas, R., Mukherjee, S., Pálfi, M., de Souza, R. S., & et al. 2022, MNRAS, 514, 1403. eprint: 2110.06184

Reichherzer, P., Schüssler, F., Lefranc, V., Becker Tjus, J., Mourier, J., & Alkan, A. K. 2023, Galaxies, 11, 22. eprint: 2212.00805

Reichherzer, P., Schüssler, F., Lefranc, V., Yusafzai, A., Alkan, A. K., Ashkar, H., & Becker Tjus, J. 2021, ApJS, 256, 5. eprint: 2109.01672

Schüssler, F., Ashkar, H., Seglar-Arroyo, M., et al. `https://github.com/astro-transients/tilepy`

Seglar-Arroyo, M., Ashkar, H., de Bony de Lavergne, M., & Schüssler, F. 2024, ApJS, 274, 1. eprint: 2407.18076

Astronomical Data Analysis Software and Systems XXXV
*ASP Conference Series, Vol. **VOL** Compiled: Tuesday 30th June, 2026@ 01:30*
K. Polsterer, D. Bohlender, S. Gwyn, A. Hincks and P. Teuben, eds.
*© **YEAR** Astronomical Society of the Pacific*

ZZZDEBUG!123.tex
Avila, B. C.|textbf
Bisero, S.
Costa, M.
Ciric, A.
Jaroschewski, I.
Kiendrébéogo, W.
Schussler, F.

Astro-COLIBRI: An Innovative Platform for Real-Time Multi-Messenger Astrophysics

Bernardo Cornejo Avila,[1] S. Bisero,[1] M. Costa,[1] A. Ciric,[1] I. Jaroschewski,[1] W. Kiendrébéogo,[1] and F. Schussler[1]

[1]*IRFU, CEA, Université Paris-Saclay, F-91191 Gif-sur-Yvette, France;*
astro.colibri@gmail.com

Abstract. The discovery of transient phenomena, such as supernovae, novae, Fast Radio Bursts (FRBs), Gamma-Ray Bursts (GRBs), and stellar flares, together with the emergence of new cosmic messengers like high-energy neutrinos and Gravitational Waves (GWs), has revolutionized astrophysics in recent years. To fully exploit the scientific potential of multi-messenger and multi-wavelength follow-up observations, as well as serendipitous detections, researchers need a tool capable of rapidly compiling and contextualizing essential information for every new event. We present Astro-COLIBRI, an advanced platform designed to meet this challenge.

Astro-COLIBRI is a comprehensive platform that combines a public RESTful API, real-time databases, a cloud-based alert system, and user-friendly interfaces including a website and mobile app for iOS and Android. It ingests alerts from multiple sources in real time, applies user-defined filters, and situates each event within its multi-messenger and multi-wavelength context. The platform provides clear data visualization, concise summaries of key event properties, and evaluations of observing conditions across a wide network of observatories worldwide. We here detail the architecture of Astro-COLIBRI, from the data pipelines that manage real-time alert ingestion and processing to the design of the RESTful API, which enables seamless integration with other astronomical software and services.

1. Introduction

Time-domain astrophysics is a rapidly growing field dedicated to the study of transient phenomena. The increasing interest in well-established science cases, such as supernovae, GRBs, or active galactic nuclei (AGN) flares, together with the discovery of new phenomena such as FRBs or GWs detections, has led to a significant rise in the number of events of interest over increasingly shorter timescales. In parallel, the continuous growth of infrastructure dedicated to transient astrophysics has expanded the amount of information available for each event.

The current observational strategy is based on two types of infrastructure. *All-sky* observatories continuously monitor the night sky and issue public alerts when an event is detected. These alerts are distributed through various brokers (GCN[1], Fink[2], TNS[3]) and received by *small field-of-view* observatories, which may or may not perform follow-up observations. The decision to follow up on an alert can be automatic (e.g.

[1]https://gcn.nasa.gov/
[2]https://fink-broker.org/
[3]https://www.wis-tns.org/

Hoischen et al. (2022), Collins et al. (2025)) or manual, depending on the procedures defined by each collaboration and on each different science case. This approach enables a comprehensive characterization of the source by combining multi-wavelength and multi-messenger data, but relies on efficient communication in a field where the volume of information and the number of tools and platforms are growing rapidly.

In this context, Astro-COLIBRI is a state-of-the-art tool that gathers all of the information necessary for the follow-up of a transient event in a user-friendly interface, available through both a web and mobile application. The platform continuously listens to public alert streams, filters them according to user needs, and places them into a multi-wavelength and multi-messenger context. Users are notified in real time through push notifications, enabling a fast and coordinated response to events of interest.

2. Astro-COLIBRI

2.1. Back-end architecture

Astro-COLIBRI combines a public RESTful API, real-time databases, a cloud-alert system, and user-friendly clients, including a web interface and a mobile application. Integration of these components is achieved through a modular architecture that follows a well-defined pipeline structured into successive processing steps.

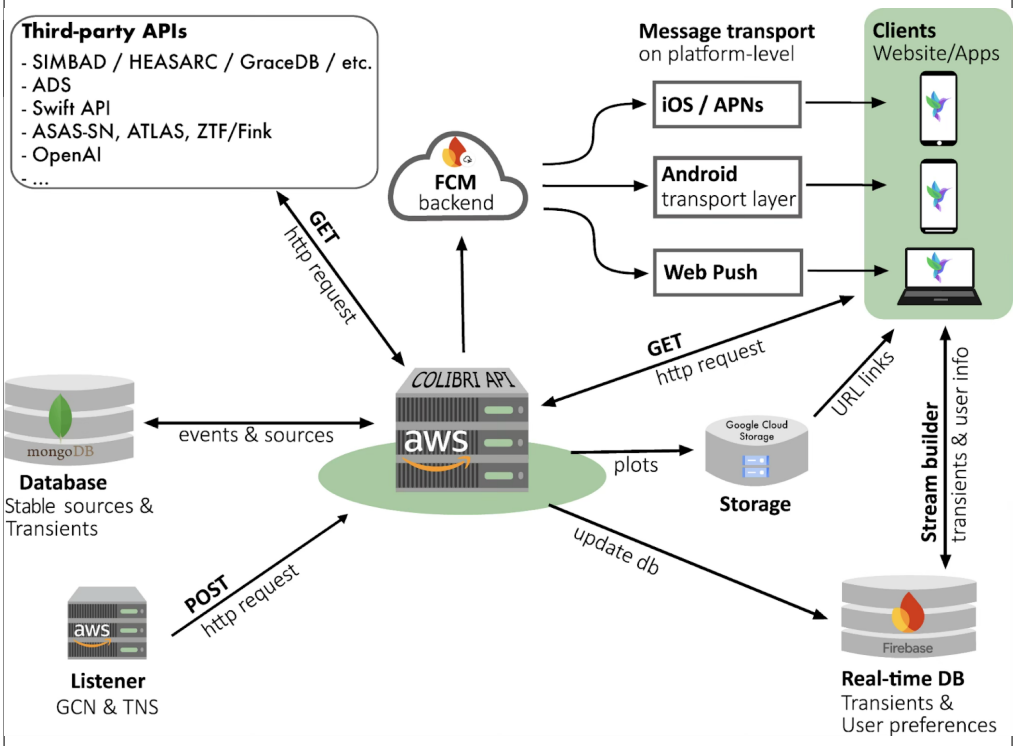


Figure 1. Architecture of Astro-COLIBRI illustrating the main system modules and their interactions, as described in detail in the text. Components highlighted in green are public interfaces. Inspired by Reichherzer et al. (2021).

1. *RESTful API*: Fully cloud-based API developed in Python using the Flask framework and running on a dedicated AWS server. It acts as the central layer of the platform, managing internal functionalities, coordinating communication between subsystems, and providing controlled access to data. It is built around a set of endpoints that handle event processing, database interactions and user requests. The API is also responsible for event-specific plot production and the communication with external third-party services. The majority of endpoints are publicly accessible and fully documented[4], with only database-modifying operations being restricted.

2. *Listener*: Continuously running on a dedicated AWS server, the event listener handles different alert streams, including those distributed by GCN, TNS, and other platforms providing VOEvent-formatted alerts. The latter are handled via the comet broker or Kafka streams, while GCN JSON notices are transmitted exclusively via Kafka. GCN circulars, TNS notifications, and email-based alerts are ingested using custom Python parsers. Incoming alerts are translated into a unified JSON-based Astro-COLIBRI format. If the trigger ID already exists in our database, the alert is deemed an update. The event is then forwarded to the API via POST requests for further processing.

3. *External APIs*: Before being stored, events get enriched using information from external platforms. This includes links to external pages (e.g. Simbad, GraceDB), public photometric data for optical transients (e.g. from ATLAS, ZTF, ASAS-SN), and public images provided by the observatories, such as IceCube neutrino track reconstruction or Fermi-GBM light curves. Additional figures may be generated to place events into their archival context. An on-demand query is used for some external services, such as the tiling and scheduling tool for poorly localized events `tilepy` (Seglar-Arroyo et al. (2024), Schussler et al. (2026)).

4. *Dual Database*: Astro-COLIBRI uses a dual-database architecture optimized for different requirements. Event data are stored as JSON-structured documents organized in collections, allowing flexible handling of heterogeneous and nested data structures. The primary database is hosted on MongoDB and serves as the backend for event processing, historical storage, and user queries, such as cone searches. In parallel, a real-time database hosted in Firebase provides low-latency data streams to all connected clients. It enables instantaneous propagation of new events and updates the interfaces without requiring page reloads, ensuring minimal cadence in the communication of a new alert following its publication. Both databases are continuously updated by the API.

5. *Push notifications*: User notifications are distributed through Firebase Cloud Messaging (FCM). When a new event or update fulfills the criteria of a predefined notification stream, a push notification is distributed to users subscribed to that stream.

2.2. Front-end

This comprehensive structure comes together in a user-friendly front-end, built using the Flutter framework, available as a web interface (`https://astro-colibri.com/`)

[4] `www.astro-colibri.science/apidoc`

and as a mobile application for Android and iOS. These clients provide easy access to the latest events, along with filters allowing users to select only those relevant to their interests. Information about each event and its potential follow-up is condensed and easily accessible. Events are visually represented in a customizable skymap.

Main Astro-COLIBRI functionalities, like cone search, event search or visibility plots, are described in detail in Reichherzer et al. (2023). New features include a photometric summary of optical sources, additional event types along with associated filter and notification capabilities, collaboration-specific ToO submission, among many other improvements.

3. Outlook

The field of multi-messenger, multi-wavelength time domain astrophysics is a constantly evolving environment. Ever-changing observation conditions, the commissioning of new facilities, and the discovery of new events all require an adaptive tool capable of responding to this rapid evolution. Built as a modular platform, Astro-COLIBRI allows for the easy implementation of updates and plug-ins, reacting promptly to emerging needs. New features and improvements are constantly being implemented to maximize the information provided to the user for each event. Finally, we are an active and accessible development team, continuously attentive to user feedback, which plays a key role in helping us build a better tool for the community.

Acknowledgments

The authors acknowledge the support of the French Agence Nationale de la Recherche (ANR) via the MOTS project (reference ANR-22-CE31-0012). This work is also supported by the European Union's Horizon Europe Research and Innovation program under the ACME project (grant agreement n. 101131928).

References

Collins, T., Egberts, K., Steppa, C., Oya, I., & Neise, D. 2025, On the Transients Handler for the Cherenkov Telescope Array Observatory. Version Number: 2, URL https://arxiv.org/abs/2508.20726

Hoischen, C., Füßling, M., Ohm, S., Balzer, A., Ashkar, H., Bernlöhr, K., Hofverberg, P., Holch, T. L., Murach, T., Prokoph, H., Schüssler, F., Zhu, S. J., Berge, D., Egberts, K., & Stegmann, C. 2022, Astronomy & Astrophysics, 666, A119. URL https://www.aanda.org/10.1051/0004-6361/202243092

Reichherzer, P., Schüssler, F., Lefranc, V., Becker Tjus, J., Mourier, J., & Alkan, A. K. 2023, Galaxies, 11, 22. URL https://www.mdpi.com/2075-4434/11/1/22

Reichherzer, P., Schüssler, F., Lefranc, V., Yusafzai, A., Alkan, A. K., Ashkar, H., & Becker Tjus, J. 2021, The Astrophysical Journal Supplement Series, 256, 5. URL https://iopscience.iop.org/article/10.3847/1538-4365/ac1517

Schussler, F., Ashkar, H., Kindrébéogo, W., Seglar-Arroyo, M., De Bony, M., Berti, A., Ruiz-Velasco, E., & Le Montagner, R. 2026, in ADASS XXXV, edited by TBD (San Francisco: ASP), vol. TBD of ASP Conf. Ser., TBD

Seglar-Arroyo, M., Ashkar, H., De Bony De Lavergne, M., & Schüssler, F. 2024, The Astrophysical Journal Supplement Series, 274, 1. URL https://iopscience.iop.org/article/10.3847/1538-4365/ad5bde

Astronomical Data Analysis Software and Systems XXXV
*ASP Conference Series, Vol. **VOL** Compiled: Tuesday 30th June, 2026@ 01:30*
K. Polsterer, D. Bohlender, S. Gwyn, A. Hincks and P. Teuben, eds.
*© **YEAR** Astronomical Society of the Pacific*

ZZZDEBUG!128.tex
Cardiel, N.|textbf
Pascual, S.
Chillarón-Víctor, M.
Cabello, C.
Gallego, J.
Zamorano, J.
Ceballos, M.T.
software!Data reduction
packages!Teareduce
computer
languages!Python

Teareduce: a Python package with utilities for teaching reduction techniques in Astronomy

Nicolás Cardiel,[1,2] Sergio Pascual,[1,2] María Chillarón-Víctor,[1,2] Cristina Cabello,[1,2] Jesús Gallego,[1,2] Jaime Zamorano,[1] and María Teresa Ceballos[3]

[1]*Departamento de Física de la Tierra y Astrofísica, Facultad CC. Físicas, Universidad Complutense de Madrid, E-28040 Madrid, Spain;* cardiel@ucm.es

[2]*Instituto de Física de Partículas y del Cosmos, IPARCOS, Facultad CC. Físicas, Universidad Complutense de Madrid, E-28040 Madrid, Spain*

[3]*Instituto de Física de Cantabria, CSIC-Universidad de Cantabria, E-39005 Santander, Spain*

Abstract. The Python package `teareduce` has been developed to support teaching activities related to the reduction of astronomical data. Specifically, it serves as instructional material for students participating in practical classes on the processing of astronomical images acquired with various instruments and telescopes. These classes are part of the course Experimental Techniques in Astrophysics, which belongs to the Master's Degree in Astrophysics at the Complutense University of Madrid. The code is publicly available on GitHub, accompanied by a documentation page that includes Jupyter notebooks demonstrating the use of its various classes and functions.

1. Introduction

Python is widely used in the scientific community and offers a powerful ecosystem for astronomical data processing. `Astropy` and affiliated packages provide ready-to-use tools for tasks like handling FITS files, performing calibrations, and analyzing photometric data, which significantly simplifies the image reduction workflow. Additionally, Python integrates seamlessly with other scientific packages (e.g., `NumPy`, `SciPy`, `Matplotlib`) to enhance data manipulation and visualization. Its cross-platform compatibility with Windows, macOS, and Linux further ensures flexibility and accessibility for researchers working in different computing environments, making Python an ideal choice for astronomical image reduction.

2. Teaching reduction techniques in the classroom

As part of the course *Técnicas Experimentales en Astrofísica* (TEA; Experimental Techniques in Astrophysics), included in the Master's Degree in Astrophysics at the Universidad Complutense de Madrid (Spain), students prepare observing proposals, carry out astronomical observations using the CAFOS instrument installed on the 2.2 m telescope at the Calar Alto Observatory (Almería, Spain; see Fig. 1), reduce the acquired images, and analyse the results.



Figure 1. *Left:* the 2.2m Telescope at the Calar Alto Observatory. *Center:* students listening to explanations from the observatory staff on the use of the CAFOS instrument. *Right:* students and faculty of the Master's program conducting observations from the control room of the 2.2m Telescope.

To support the data reduction and analysis process, we have developed the Python package `teareduce`[1]. Its main goal of is to serve as an educational tool.

`Teareduce` is not intended as a general-purpose image reduction tool. Rather, it includes a set of specific operations needed for certain steps in the traditional astronomical image reduction workflow. At the time of its creation, these features were not available in more established packages such as `ccdproc`. Students can examine the Python code and introduce modifications in order to gain a deeper understanding of the operations performed during the astronomical image reduction process. The package also introduces alternative approaches to some tasks, which have proven to be more practical for use in Master's-level instruction. Some examples are shown below.

2.1. Auxiliary classes for image slicing

Working with astronomical images in FITS (Flexible Image Transport System) format is a fundamental task in modern astronomy. However, one of the most common and persistent sources of programming errors when handling FITS files in Python stems from the different indexing and slicing conventions used across various systems. These differences, rooted in historical computing decisions, continue to confuse astronomers and programmers alike, leading to subtle bugs that can compromise scientific results.

The `teareduce` classes `SliceRegion1D`, `SliceRegion2D`, and `SliceRegion3D` allow the users to explicitly define the slice region following either FITS or Python indexing convention. Once an object is instantiated, it can also be used with either the FITS or Python convention by simply accessing the corresponding attribute: `.fits` or `.python`.

2.2. Wavelength calibration

The C-distortion correction and the wavelength calibration of astronomical spectra are very important steps in the reduction of astronomical spectra. The `teareduce` package provides the `TeaWaveCalibration` class, which enables the following steps to be performed: interactive identification of arc lines, automatic detection of line peaks across all spectra in a 2D image, and computation of polynomial corrections to perform

[1] `https://nicocardiel.github.io/teareduce-cookbook/intro.html`

the C-distortion correction and the wavelength calibration (see Fig. 2). The resulting calibration can be applied both to `Numpy` arrays and to `CCDData`[2] objects.

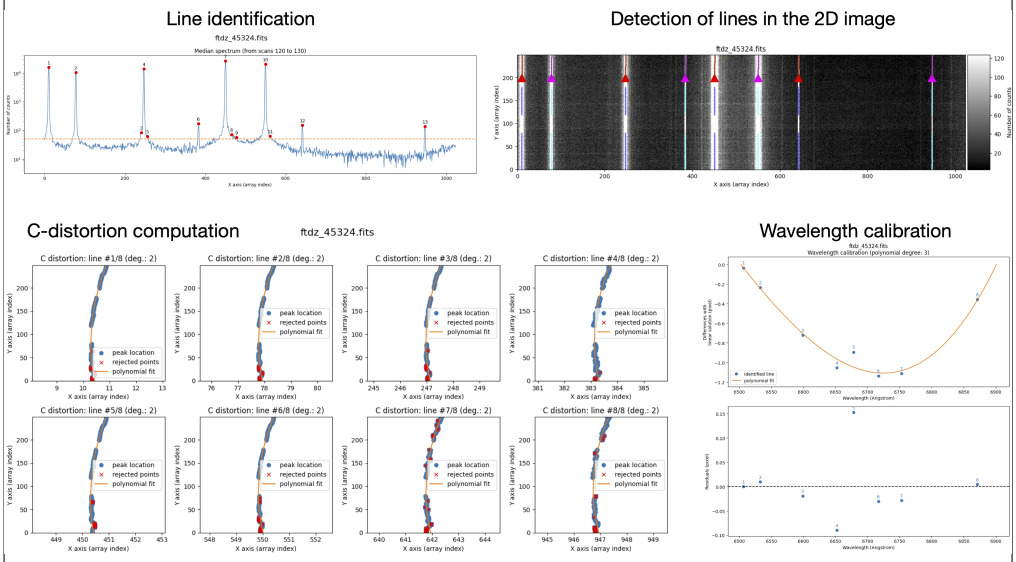


Figure 2. Different steps carried out by the `TeaWaveCalibration` class.

2.3. Adaptive spline fit

There are many situations in which it is useful to perform a smooth fit to a series of points, but a polynomial fit does not provide the necessary flexibility. In such cases, spline fitting is often used.

To facilitate this task, the class `AdaptiveLSQUnivariateSpline` allows performing this type of fit without having to predefine the locations of the knots (the points where the different polynomial segments join to create a continuous and smooth curve), requiring only the specification of the number of intermediate knots to use (see e.g., Cardiel 2009). Applying a numerical minimisation process allows the fit to automatically adjust the knot positions in order to achieve a good match to the data.

2.4. Cosmic ray removal

The `teareduce` package includes an auxiliary program called `tea-cleanest`, which enables the interactive cleaning of cosmic rays in individual exposures (see Fig. 3). This code is inspired by the `cleanest` code (Cardiel 2020)[3], although the approach to detecting cosmic rays differs. In particular, this program uses the L.A. Cosmic algorithm (van Dokkum 2001) to identify pixels suspected of being affected by cosmic-ray hits.

At the time of writing, other algorithms, such as PyCosmic (Husemann et al. 2012), deepCR (Zhang & Bloom 2020) and Cosmic-CoNN (Xu et al. 2023), are being incorporated into this tool.

[2] `https://docs.astropy.org/en/stable/nddata/ccddata.html`
[3] `https://cleanest.readthedocs.io/`

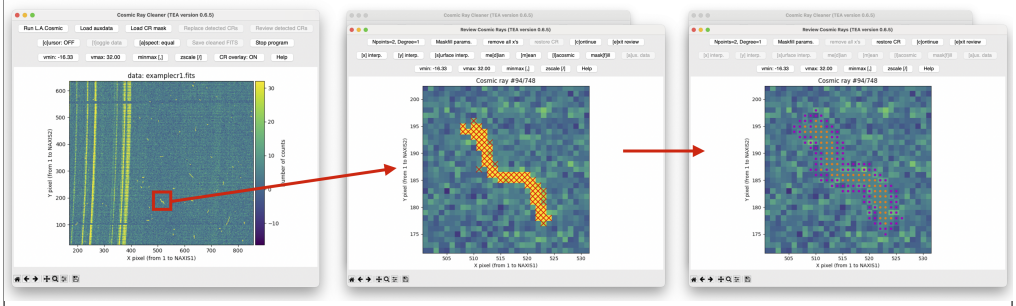


Figure 3. Example of cosmic ray detection and correction: original image with multiple cosmic-ray hits (*left*), identification of a specific cosmic ray with pre-flagged pixels marked by red x's allowing manual selection/deselection (*center*), and corrected image after interpolation with interpolated pixels marked by x's and pixels used for interpolation indicated by magenta dots (*right*).

`Teareduce` also provides an alternative method, the function `cr2images`, for automatically removing cosmic rays in cases where two equivalent exposures are available. In such situations, the cosmic-ray pixels in each exposure can be replaced with the signal from the other exposure. This function also allows the users to define regions to be cleaned or regions to be excluded from the cleaning process.

Acknowledgments. We thank the support of the projects PID2021-123417OB-I00 and PID2022-138621NB-I00, funded by MCIN/AEI/10.13039/501100011033/FEDER, EU, and PCI2022-135023-2, funded by MCIN/AEI/10.13039/501100011033 and by the European Union 'NextGenerationEU'/PRTR. This work made use of `astropy` (Astropy Collaboration et al. 2013, 2018), `numpy` (Harris et al. 2020), `scipy` (Virtanen et al. 2020), `ccdproc` (Craig et al. 2017), and `matplotlib` (Hunter 2007).

References

Astropy Collaboration, Price-Whelan, A. M., et al. 2018, AJ, 156, 123. eprint: 1801.02634
Astropy Collaboration, Robitaille, T. P., et al. 2013, A&A, 558, A33. eprint: 1307.6212
Cardiel, N. 2009, MNRAS, 396, 680. eprint: 0903.2068
— 2020, in Astronomical Data Analysis Software and Systems XXVII, edited by P. Ballester, J. Ibsen, M. Solar, & K. Shortridge, vol. 522 of Astronomical Society of the Pacific Conference Series, 723
Craig, M., et al. 2017, astropy/ccdproc: v1.3.0.post1. URL `https://doi.org/10.5281/zenodo.1069648`
Harris, C. R., et al. 2020, Nat, 585, 357. eprint: 2006.10256
Hunter, J. D. 2007, Computing in Science & Engineering, 9, 90
Husemann, B., Kamann, S., Sandin, C., Sánchez, S. F., García-Benito, R., & Mast, D. 2012, A&A, 545, A137. eprint: 1208.1696
van Dokkum, P. G. 2001, PASP, 113, 1420. eprint: astro-ph/0108003
Virtanen, P., et al. 2020, Nature Medicine, 17, 261. eprint: 1907.10121
Xu, C., McCully, C., Dong, B., Howell, D. A., & Sen, P. 2023, ApJ, 942, 73. eprint: 2106.14922
Zhang, K., & Bloom, J. S. 2020, ApJ, 889, 24. eprint: 1907.09500

Astronomical Data Analysis Software and Systems XXXV
*ASP Conference Series, Vol. **VOL** Compiled: Tuesday 30th June, 2026@ 01:30*
K. Polsterer, D. Bohlender, S. Gwyn, A. Hincks and P. Teuben, eds.
*© **YEAR** Astronomical Society of the Pacific*

ZZZDEBUG!129.tex
Yang, H.|textbf
Cui, C.
Li, S.
FOOBAR!conference!ADASS 2019
FOOBAR!organisations!ASP
FOOBAR! ascl:1101.010

Platform, Participation, and Progress:China's Recent Advances in Astronomical Citizen Science

Hanxi Yang,[1] Chenzhou Cui,[2] and Shanshan Li,[3]

[1]*National Astronomical Observatories, Chinese Academy of Science, Beijing, China;* yanghanxi@nao.cas.cn

[23]*National Astronomical Observatories, Chinese Academy of Science, Beijing, China*

Abstract. Astronomy has entered a data-intensive era, offering unprecedented opportunities for public involvement in scientific discovery. In China, the China-VO team of the National Astronomical Data Center (NADC) has played a pioneering role in the systematic development and institutionalization of citizen science. Starting with early outreach milestones, such as organizing the "Yangtze River Solar Eclipse" multi-channel joint observation live broadcast and establishing the "Universe Portal" website, China-VO has built significant influence and a strong brand within the amateur astronomy community. After 20 years of development, astronomical citizen science in China has formed a comprehensive framework driven by three core dimensions: Platform (unified infrastructure), Participation (diverse mechanisms and project-based engagement), and Progress (strategic milestones and scientific outcomes). This triad of forces has created a virtuous cycle that accelerates research and enhances public scientific literacy. This paper systematically reviews China's journey, from early community building to the 2025 launch of a unified platform and the first Citizen Science Summit in Astronomy (CSSA), thereby offering a sustainable model for open, collaborative science in the era of big data.

1. Platform: Building an Infrastructure for Inclusive Science

1.1. From a Success Case to an Integrated Ecosystem

The platform is designed to be more than just a task interface; it is a comprehensive ecosystem. Its core features include:

1.2. A Comprehensive Hub for Research and Education

The foundation of the China-VO Citizen Science Platform was laid by the long-standing success of the Popular Supernova Project (PSP), which launched in 2015 as a collaborative effort between China-VO and the Xingming Observatory in Xinjiang, PSP demonstrated the immense potential of public participation in frontline research. Over a decade, the project engaged nearly 20,000 participants and successfully identified 212 supernova candidates. PSP's data enabled a Nature publication, significantly elevating the scientific value and legitimacy of citizen science. Seeing this proven track record, an increasing number of professional scientists chose to partner with China-VO to launch diverse projects, such as galaxy classification and spectral inspection. To effectively

manage these growing initiatives and lower the barrier for new users, the National Astronomical Data Center (NADC) officially established the unified China-VO Citizen Science Platform in 2025.

- Diverse Project Portals: Categorized by difficulty and scientific topic, allowing participants to choose tasks that match their interests.
- Educational Resources: The platform provides detailed explanations of citizen science, introduces the specific missions of each project, and showcases scientific outcomes to inspire and educate the public.
- Learning and Community: By integrating multimedia tutorials and discussion forums, the platform fosters a sense of community and helps participants transition from novices to capable contributors.

2. Participation: A Rich and Diverse Ecosystem

With the establishment of this platform, China's astronomical citizen science has seen a significant increase in the number and variety of projects, offering users a wide range of choices. These projects are primarily divided into two major categories based on the nature of the tasks, allowing participants to freely choose based on their interests and skill levels:

- Object Identification includes: Popular Supernova Project (PSP), SpecZoo, GalaxyCircus, GalaxyMaze, LensFinder, and Radio Galaxy Zoo: EMU.
- Event Reporting includes: Fireball reporting system, Exoplanet-2025, and Celestial constellation in Folk Culture.

This high degree of project richness provides users with more options, allowing them to freely choose domains that interest them. This diversity not only lowers the barrier to entry but also creates a powerful multiplier effect, greatly promoting the flourishing development of astronomical citizen science in China.

3. Progress: From 20 Years of Practice to Strategic Maturation

The Progress of astronomical citizen science in China is characterized by a transition from early spontaneous efforts to a unified, strategic framework. After twenty years of rich practical experience the field reached a critical tipping point. The convening of the first Citizen Science Summit in Astronomy (CSSA) in 2025 was, therefore, perfectly timed to transform two decades of empirical success into a coherent strategic vision.

The CSSA 2025 served as a symbolic event, synthesizing twenty years of experience into a structured ecosystem. The summit established a strategic consensus within the astronomical community on three aspects:

- WHAT | Defining the Field: Establishing a formal, consensus-based definition of citizen science as a collaborative research model between professionals and the public tailored to the Chinese context.



Figure 1. The frontpage of China-VO Citizen Science Platform.

- WHY | Demonstrating Value : Documenting the multifaceted impact of these initiatives, from accelerating high-level research to enhancing public scientific literacy. HOW | Enabling with Technology: Exploring a "human-AI collaboration" paradigm where cutting-edge technologies like AI and Big Data are integrated to make projects more efficient and engaging.
- WHERE TO | The 2025 Road map: Laying out a strategic plan focused on four key pillars: Developer Community, Project R&D, Education & Support, and Standardization.

4. Summary

Looking toward the future, as the field moves beyond data processing to deeper scientific involvement and embraces international collaboration, this framework will continue to evolve. By synthesizing twenty years of practice into a strategic roadmap, China's advances offer a valuable "China Model" for the global community, contributing significantly to the future of open science at large.

5. Images



Figure 2. Projects of the Object Identification type



Figure 3. Projects of the event reporting type

Acknowledgments. This work is supported by the National Key R&D Program of China (2022YFF0711500), National Natural Science Foundation of China (NSFC)(12573111, 12403102, 12373110, 12273077, 12103070), Strategic Priority Research Program of the Chinese Academy of Sciences (XDB0550101). Data resources are supported by China National Astronomical Data Center (NADC), CAS Astronomical Data Center and Chinese Virtual Observatory (China-VO). This work is supported by Astronomical Big Data Joint Research Center, co-founded by National Astronomical Observatories, Chinese Academy of Sciences and Alibaba Cloud.

Astronomical Data Analysis Software and Systems XXXV
*ASP Conference Series, Vol. **VOL** Compiled: Tuesday 30th June, 2026@ 01:30*
K. Polsterer, D. Bohlender, S. Gwyn, A. Hincks and P. Teuben, eds.
*© **YEAR** Astronomical Society of the Pacific*

ZZZDEBUG!130.tex
Findeisen, K.|textbf
Lim, K-T.
Speck, D.
Chiang, H-F.
Howard, E. L.
Sullivan, I. S.
Bellm, E. C.
data!processing!automated
observatories!ground-based
pipelines!Rubin Prompt Processing
surveys!LSST

The Vera C. Rubin Observatory Prompt Processing System

Krzysztof Findeisen,[1] Kian-Tat Lim,[2] Dan Speck,[3] Hsin-Fang Chiang,[2] Erin Leigh Howard,[1] Ian S. Sullivan,[1] and Eric C. Bellm[1]
[1]*University of Washington, Seattle, WA, USA;* `kfindeis@uw.edu`

[2]*SLAC National Accelerator Laboratory, Menlo Park, CA, USA*

[3]*Burwood Group, Oak Brook, IL, USA*

Abstract. Vera C. Rubin Observatory's Prompt Processing system will automatically process 10 TB of raw images to produce up to 10 million transient alerts per night. We summarize how Prompt Processing meets its throughput, latency, and reliability requirements and presents results from Rubin Observatory Commissioning.

1. Introduction

The NSF-DOE Vera C. Rubin Observatory's Legacy Survey of Space and Time (LSST, Ivezić et al. 2019) will generate unprecedented data volumes. The LSST Camera (Roodman et al. 2024), with 189 4K×4K detectors, produces 8 GB of uncompressed data every 39 s and 10 TB per night. LSST data will be processed in two modes: annual Data Releases and real-time Prompt Processing. Prompt Processing handles images immediately, identifying new, variable, or moving sources and generating up to 10,000 alerts per visit and 10 million per night, all within 60-120 s of shutter close.

This paper describes how the Prompt Processing system (at `https://github.com/lsst-dm/prompt_processing/`) handles a large data volume to produce alerts with low latency, with little downtime or human intervention, while executing the Alert Production pipeline (Rubin Observatory Science Pipelines Developers 2025).

2. Prompt Processing Design

2.1. Key Components

To flexibly scale with changes to observing cadence, processing time, and other variables, Lim (2023) proposed a task-based architecture with one task for each detector of each visit (one "visit-detector"). The tasks are submitted to a worker pool running on a Kubernetes cluster, each worker processing its task independently in an individual workspace. Independent execution maximizes performance, per Amdahl's law, while a cloud-native worker pool benefits from existing solutions for autoscaling, error recovery, metrics, etc.

While Lim (2023) originally implemented Prompt Processing with the Knative Serverless Platform, we migrated to Kubernetes Event-Driven Autoscaling (KEDA) in

early 2025 after finding Knative is ill-suited for long-running tasks. The current implementation can scale up to 1700 Kubernetes pods, limited by cluster memory. Each pod handles multiple tasks over its lifetime, amortizing startup costs and allowing caching between runs. A shared Redis Streams queue tracks tasks and holds any overflow.

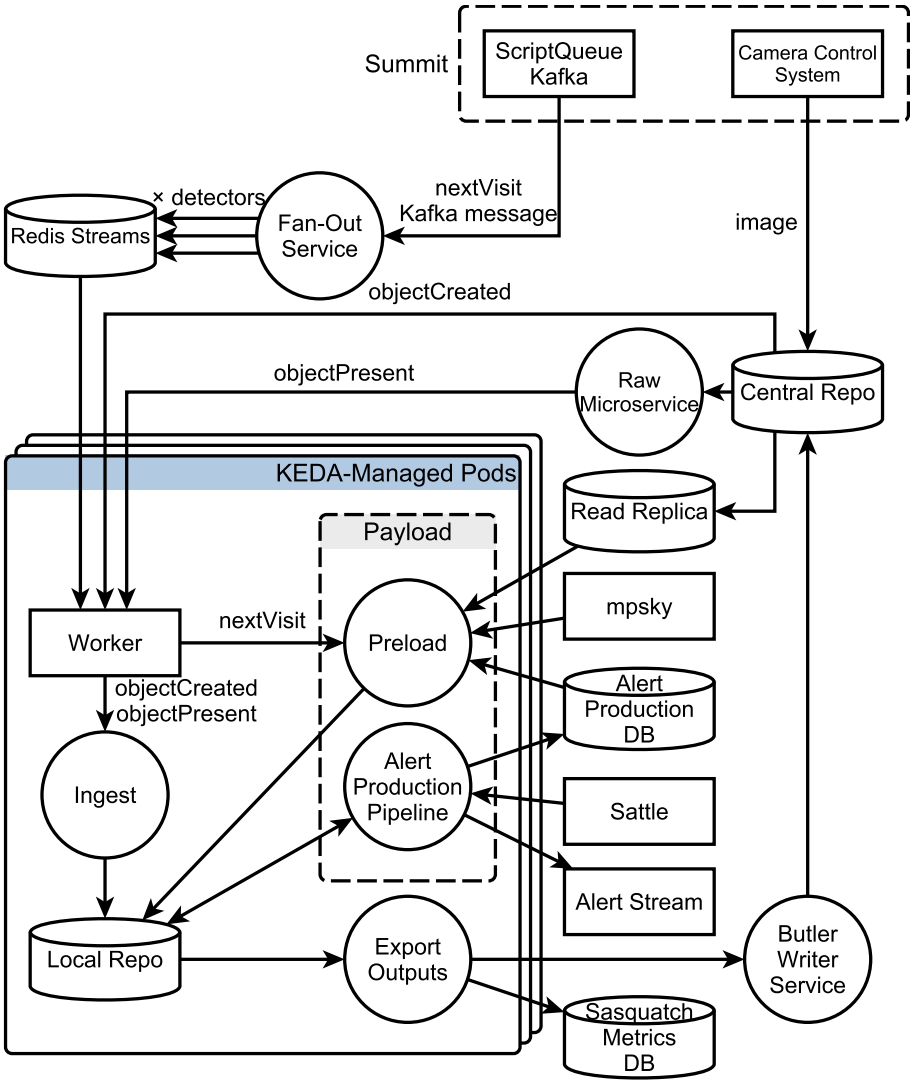


Figure 1. Data flow diagram for the Prompt Processing system, showing interaction with nextVisit messages and other Rubin Observatory systems.

While the system confines processing to independent workers as much as possible, it must interact with several shared resources, shown in Fig. 1: a central data repository (Jenness et al. 2022) to load calibrations, difference imaging templates, and other inputs, and to store pipeline products; the Cassandra Alert Production database (APDB, Salnikov & McCormick 2025) to load and update a record of the variable sky; the `mpsky` ephemeris service to identify solar system objects; and the Sattle service to vet candidate sources against known satellites.

2.2. Workflow

To minimize alert latency, visit processing begins before the image is taken. As shown in Fig. 1, the Summit facility sends a "nextVisit" message one visit or 20 s (whichever is longer) before exposure start, which a microservice converts into visit-detector tasks. On picking up a new task, each worker preloads the needed inputs from the central repository, the APDB, and `mpsky`. Preload normally finishes before the image arrives, letting pipeline execution begin immediately. Workers cache calibration data, but this is not practical for template images and other field-specific inputs.

Normally, image arrival triggers a Kafka notification that the worker picks up with minimal delay. As a backup, a microservice indexes previously received images. When either mechanism detects the awaited image, the worker ingests it into its own workspace and runs the pipeline locally. Whether the pipeline succeeds or fails, the worker exports pipeline products and metrics as a separate step.

Prompt Processing does not have a hard-coded pipeline, instead abstracting pipelines through the Rubin Observatory Middleware framework (Jenness et al. 2022). This lets the system configure different pipelines based on nextVisit metadata, as well as prioritized lists of pipelines to be attempted.

As noted in Section 2.1, avoiding shared resources or operations is essential to good parallel performance. External I/O is minimized, and restricted as much as possible to preload and export, where any delays contribute to worker load but not alert latency. There are three exceptions: the Alert Production pipeline writes to the APDB and the alert stream, which are both optimized for bulk writes, and queries the Sattle service, which is low latency. Sending alerts before cleaning up pipeline execution not only gets them to alert brokers faster, but also minimizes the risk that the alerts will be lost if another operation fails (Findeisen et al. 2025).

Pipeline failures are handled through the Middleware framework, and by default assumed to be repeatable. If a worker fails for non-algorithmic reasons, such as I/O errors or pod termination, we consider a visit-detector retriable if it has not updated the APDB (Findeisen et al. 2025). Such retries are not yet implemented with KEDA.

3. Commissioning Results

Prompt Processing was first deployed in May 2023, handling live imaging data from the LATISS instrument on the Rubin Auxiliary Telescope (NSF-DOE Vera C. Rubin Observatory 2020). It ran at increasing scale when the LSST Commissioning Camera (Stalder et al. 2022) and the LSST Camera (Roodman et al. 2024) went on sky in November 2024 and April 2025, respectively. This two and a half year commissioning process gave us experience running Prompt Processing in a variety of failure scenarios.

Prompt Processing sent 4 million alerts in 1-2 hours each night on July 1 and July 2 to Rubin's alert brokers, demonstrating the required throughput. However, the central repository and the APDB struggled to handle over a thousand simultaneous clients. Slow or timed out queries degraded latency and reliability. The Middleware team added a read-only database replica and a serialized writer service, while the Alert Production and Data Access teams streamlined APDB read and write volume. These will be tested at scale as survey-like observations resume.

From July 12 to the end of LSSTCam Commissioning, Prompt Processing handled 73% of all science images, with 22% lost when the read replica fell out of date (blocking

data access until manually reset), 3% when unexpected camera rotations invalidated preload, and 2% to other causes. These issues have since been mitigated: the read replica is more reliable, and has redundant checks and alerts to detect desynchronization earlier, while scheduler and Summit workflow improvements will ensure that images are almost always taken as originally planned.

4. Ongoing Work

We continue to identify and mitigate failure modes during the Rubin Early Optimization Phase (Guy et al. 2025), and plan to restore support for retrying transient errors in KEDA. We also need to test existing solutions to the shared-resource bottlenecks at full survey cadence, though smaller tests are encouraging. Reducing latency will benefit more from optimizing pipeline payloads than Prompt Processing itself.

There will inevitably be failures requiring human intervention, particularly if recovery involves reconciling APDB history. We plan to create a daytime processing service to handle this small number of cases.

Acknowledgments. This material is based upon work supported in part by the National Science Foundation through Cooperative Agreements AST-1258333 and AST-2241526 and Cooperative Support Agreements AST-1202910 and 2211468 managed by the Association of Universities for Research in Astronomy (AURA), and the Department of Energy under Contract No. DE-AC02-76SF00515 with the SLAC National Accelerator Laboratory managed by Stanford University. Additional Rubin Observatory funding comes from private donations, grants to universities, and in-kind support from LSST-DA Institutional Members.

References

Findeisen, K., et al. 2025, Failure Modes and Error Handling for Prompt Processing, DMTN 260, NSF-DOE Vera C. Rubin Observatory. URL https://dmtn-260.lsst.io/

Guy, L. P., et al. 2025, Rubin Observatory Plans for an Early Science Program, RTN 011, NSF-DOE Vera C. Rubin Observatory. URL https://rtn-011.lsst.io/

Ivezić, Ž., et al. 2019, ApJ, 873, 111. eprint: 0805.2366

Jenness, T., et al. 2022, in Software and Cyberinfrastructure for Astronomy VII, vol. 12189 of SPIE Conf. Ser., 1218911. eprint: 2206.14941

Lim, K.-T. 2023, Proposal and Prototype for Prompt Processing, DMTN 219, NSF-DOE Vera C. Rubin Observatory. URL https://dmtn-219.lsst.io/

NSF-DOE Vera C. Rubin Observatory 2020, LSST Atmospheric Transmission Imager and Slitless Spectrograph (LATISS). URL https://doi.org/10.71929/rubin/2571930

Roodman, A., et al. 2024, in Ground-based and Airborne Instrumentation for Astronomy X, vol. 13096 of SPIE Conf. Ser., 130961S

Rubin Observatory Science Pipelines Developers 2025, The LSST Science Pipelines Software: Optical Survey Pipeline Reduction and Analysis Environment, PSTN 019, NSF-DOE Vera C. Rubin Observatory. URL https://pstn-019.lsst.io/

Salnikov, A., & McCormick, J. 2025, Current status of APDB and PPDB implementation, DMTN 293, NSF-DOE Vera C. Rubin Observatory. URL https://dmtn-293.lsst.io/

Stalder, B., et al. 2022, in Ground-based and Airborne Instrumentation for Astronomy IX, vol. 12184 of SPIE Conf. Ser., 121840J

Astronomical Data Analysis Software and Systems XXXV
*ASP Conference Series, Vol. **VOL** Compiled: Tuesday 30th June, 2026@ 01:30*
K. Polsterer, D. Bohlender, S. Gwyn, A. Hincks and P. Teuben, eds.
*© **YEAR** Astronomical Society of the Pacific*

ZZZDEBUG!133.tex
Taghizadeh-Popp, M.|textbi
Cherinka, B.
Weijmans, A.
Brownstein, J. B.
Donor, J.
Thakar, A.
surveys!Sloan Digital Sky Survey (SDSS)
Sloan Digital Sky Survey (SDSS)
catalogues!individual!SDSS
cross-matching!services
computer languages!JavaScript
databases!query language!SQL
applications!Aladin Lite
visualisation!web-based
web!technologies
web!application
projects!SciServer Compute

New Visualization and Analysis Capabilities for SDSS DR19

Manuchehr Taghizadeh-Popp,[1] Brian Cherinka,[2] Anne-Marie Weijmans,[3] Joel R. Brownstein,[4] John Donor,[5] and Ani Thakar [1]

[1] *Johns Hopkins University, Baltimore, MD 21218, USA;* mtaghiza@jhu.edu

[2] *Space Telescope Science Institute, Baltimore, MD 21218, USA*

[3] *University of St Andrews, North Haugh, St Andrews KY16 9SS, UK*

[4] *University of Utah, Salt Lake City, UT 84112, USA*

[5] *Texas Christian University, Fort Worth, TX 76129, USA*

Abstract.
The latest Sloan Digital Sky Survey (SDSS) Data Release 19 (DR19) comes with a new suite of tools to help astronomers and students visualize and analyze the vast richness of this dataset. In this demo we showcase several of these tools, starting with programmatic data analysis with Python in Jupyter Notebook tutorials on the SciServer science platform. We then highlight several interactive tools, including a) the Zora web application - a modern and reactive interface for searching SDSS data, exploring observed target metadata, and visualizing or accessing spectroscopic data; b) the redesigned SkyServer Navigate tool for browsing through SDSS source catalogs, footprints, and geometries on an interactive sky map; and c) the SQLxMatch tool (also on SkyServer) for running on-the-fly cross-matches of SDSS objects against more than 50 other astronomical catalogs.

1. Introduction

For more than 25 years, the SDSS collaboration has been making groundbreaking observations and releasing valuable astronomical data sets to the public on an almost annual basis. At the same time, it has also managed to build the cyber-infrastructure needed to provide data access, analysis, and visualization capabilities to a wide audience, ranging from professional astronomers to high school students. Now that SDSS is in its 5th incarnation(Kollmeier et al. 2026) and information technologies have evolved and improved, the latest SDSS DR19 (SDSS Collaboration et al. 2025) brings a new suite of software, applications and tools that help users visualize and analyze the latest data. The purpose of this demo is to introduce these new ways of working with the latest SDSS data sets.

2. Data analysis with Jupyter Notebook tutorials on the SciServer platform

SDSS leverages the SciServer science platform (Taghizadeh-Popp et al. 2020) to host all data releases and provide users with a collaborative environment for programmatic

astronomical analysis with SDSS data in both interactive and batch mode. In DR19, we have developed a suite of Jupyter Notebook tutorials in Python that showcase multiple science use cases using the latest survey mapper data available on the SciServer file system, from the Black Hole Mapper, Milky Way Mapper, and Local Volume Mapper (SDSS Collaboration et al. 2025). Users can launch these Notebooks in Jupyter Lab sessions in SciServer within a containerized Docker environment. This environment contains libraries and packages required for astronomical data analysis, such as `sdss_access`, `semaphore`, `astropy`, and `specutils`. For quick dissemination and outreach support, we have made these Notebooks public in the SDSS Github repository (`www.github.com/sdss/dr19_tutorials`). A Notebook snippet can be seen on the left panel of Figure 1.

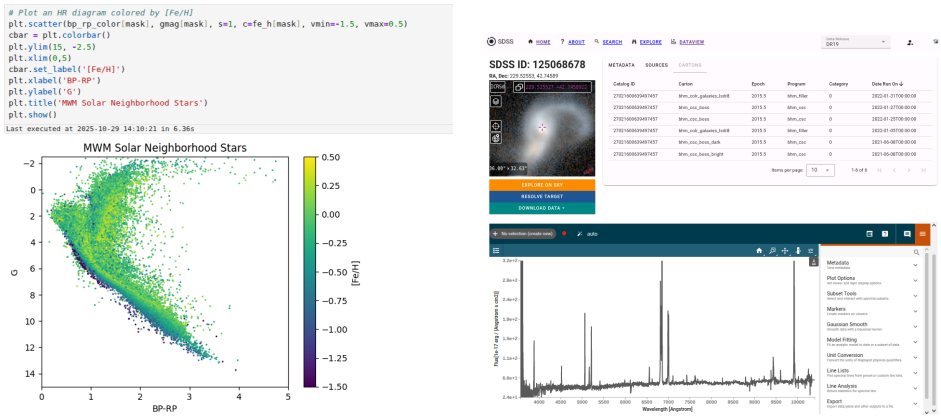


Figure 1. Data access modalities. *Left:* Programmatic access with Jupyter Notebooks in SciServer. Shown is a Notebook snippet that includes a code cell and a created figure. *Right:* Interactive access with the Zora web application.

3. Data exploration and visualization with Zora and Valis web tools

The new Zora application (`https://dr19.sdss.org/zora/`) is a public and modern web UI - built around the lightweight, reactive, and modular VueJs framework - that can be used to quickly access and interactively search specific SDSS objects and explore observed target metadata, which complements the programmatically-oriented data access with Jupyter Notebooks in SciServer. Specifically, Zora users can search objects by `sdss_id` or alternate ids (e.g. 2MASS or Gaia), and inspect a single source target. Zora also incorporates a spectrum viewer, based on JDaviz (JDADF Developers et al. 2025), that can be used for interactive visualization and analysis of spectral data. Zora also has a Sky Viewer based on Aladin-Lite (Baumann et al. 2022), used to explore the SDSS footprint, display SDSS summary catalogs on the sky, and perform cone searches on the sky via a search form. Another new interactive feature is Zora's Dataview Dashboard, based on the Vaex and Solara frameworks, which can be used to explore and visualize the multidimensional space of catalog parameters. A screenshot of Zora can be seen in the right panel of Figure 1.

For programmatic API access on the web, SDSS data is served by Valis (`https://api.sdss.org/valis/docs/`) - a public web service that exposes a REST API and

is built with the FastAPI Python framework. One of the main users of Valis is Zora, which makes http calls to Valis in order to retrieve the data it displays to astronomers, students, or the general public. Services provided by Valis include cone searches, target id searches, target metadata or pipeline parameter lookups, the retrieval of spectral data as JSON, download or stream of files, and retrieval of information regarding SDSS data models, releases, versions, environments, and file paths, among others.

4. Visual data exploration and multi-catalog cross-match in SkyServer

SkyServer (`https://skyserver.sdss.org`) is a legacy web UI in SDSS that provides an interface for exploring astronomical objects and launching raw and form-based SQL queries against the SDSS database. New updates in SkyServer for DR19 include fully redesigned Navigate and Cross-Match tools (Figure 2), as described next.

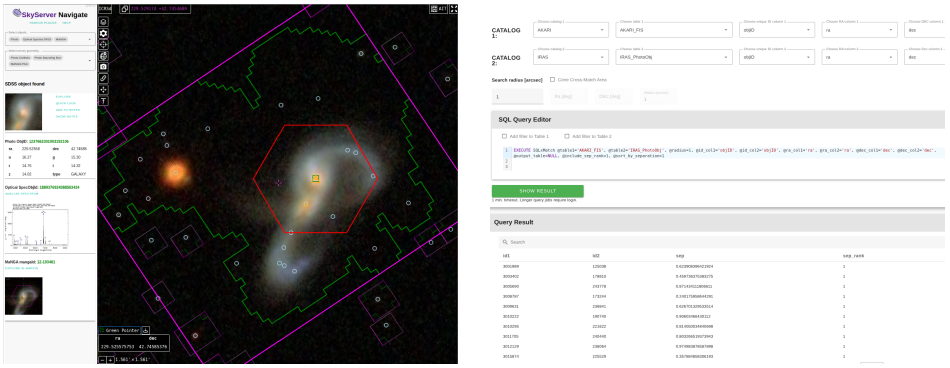


Figure 2. New updates on SkyServer. *Left:* Fully redesigned SkyServer-Navigate sky map explorer. *Right:* Screenshot of the SQLxMatch cross-match web interface

.

The newly upgraded SkyServer Navigate tool (Taghizadeh-Popp 2025) works as an interactive sky map viewer in a web browser. It presents a faster and reactive interface at all zoom levels compared to the previous version, and can be used by a wide range of audiences at different levels. Navigate is a client-side application written in the reactive Vue.js Javascript framework. Its sky map is based on the Aladin Lite viewer (Baumann et al. 2022), which allows users to display and pan over the SDSS photometric footprint, and view sky maps from multiple other surveys that span multiple wavelengths on adjustable spherical projections.

Navigate also allows users to add SDSS-specific overlays on the sky maps and interact with them. These include object catalogs represented by markers on the positions of objects observed under the mappers and surveys within SDSS. Clicking on a marker displays metadata associated with the corresponding object, such as spectra, photometric information, and links to other tools within SkyServer. We use the HiPS tiling framework (Fernique et al. 2015) to create and display object catalogs on multiple zoom levels, which can be displayed on-demand without overcrowding the map.

Overlays involving the display of polygonal or circular shapes that outline the geometries and footprints of the multiple SDSS surveys are also possible. These include outlines of spectroscopic plates and IFUs, photometric masks and CCD fields, photo-

metric outlines and bounding boxes of observed objects, and survey footprint outlines. Navigate also implements a full REST API that helps with programmatic reproducibility and shareability of particular map views between users.

The new SQLxMatch tool (Taghizadeh-Popp & Dobos 2023) allows users to run a 2-way spatial cross match or cone search on the sky between objects from SDSS and more than 50 other astronomical catalogs, by means of SQL queries that the tool helps users to construct. To this end, we have stored these catalogs and made them available in the `xmatch` database in the CasJobs application (`https://skyserver.sdss.org/casjobs`)- a legacy SDSS query workbench application to run sync and async SQL queries against SDSS databases. In turn, users can connect to the `xmatch` database in order to execute the SQLxMatch SQL stored procedure, which specifically runs the cross-match/cone search on the astronomical objects listed in the catalog tables. This procedure leverages the Zones Algorithm (Gray et al. 2007), which divides the sky into stripes across constant declination, and uses relational database algebra and B-Trees to compute a fast in-database cross-match between the catalog tables.

Users can also upload object catalogs into their personal MyDB database, and cross-match them against the public catalog tables. This cross-match is computed on-the-fly, and users can freely provide their own search radius and pre-filter catalog tables within the SQL query, so that only the results of the cross-match are returned to them. Besides CasJobs, there are two other access modes: users can either use the web form on the SkyServer "Cross-Match" page to run interactive cross-match queries, or run Jupyter Notebooks on SciServer that leverage the CasJobs REST API to programmatically run cross-match queries in interactive or batch mode.

Acknowledgments. Funding for the Sloan Digital Sky Survey V has been provided by the Alfred P. Sloan Foundation, the Heising-Simons Foundation, the National Science Foundation, and the Participating Institutions. SDSS acknowledges support and resources from the Center for High-Performance Computing at the University of Utah. SDSS telescopes are located at Apache Point Observatory, funded by the Astrophysical Research Consortium and operated by New Mexico State University, and at Las Campanas Observatory, operated by the Carnegie Institution for Science. The SDSS web site is `www.sdss.org`.

References

Baumann, M., Boch, T., Pineau, F.-X., Fernique, P., Bot, C., & Allen, M. 2022, in Astronomical Data Analysis Software and Systems XXX, edited by J. E. Ruiz, F. Pierfedereci, & P. Teuben, vol. 532 of Astronomical Society of the Pacific Conference Series, 7

Fernique, P., et al. 2015, A&A, 578, A114. eprint: 1505.02291

Gray, J., Nieto-Santisteban, M. A., & Szalay, A. S. 2007, CoRR, abs/cs/0701171. eprint: cs/0701171, URL `http://arxiv.org/abs/cs/0701171`

JDADF Developers, et al. 2025, Jdaviz. URL `https://doi.org/10.5281/zenodo.5513927`

Kollmeier, J. A., et al. 2026, AJ, 171, 52. eprint: 2507.06989

SDSS Collaboration, et al. 2025, arXiv e-prints, arXiv:2507.07093. eprint: 2507.07093

Taghizadeh-Popp, M. 2025, Skyserver-navigate: a sky map viewer for sdss and beyond. URL `https://doi.org/10.5281/zenodo.15356714`

Taghizadeh-Popp, M., & Dobos, L. 2023, Sqlxmatch: In-database spatial cross-match of astronomical catalogs. URL `https://doi.org/10.5281/zenodo.10160988`

Taghizadeh-Popp, M., et al. 2020, Astronomy and Computing, 33, 100412. eprint: 2001.08619

Astronomical Data Analysis Software and Systems XXXV
*ASP Conference Series, Vol. **VOL** Compiled: Tuesday 30th June, 2026@ 01:30*
K. Polsterer, D. Bohlender, S. Gwyn, A. Hincks and P. Teuben, eds.
*© **YEAR** Astronomical Society of the Pacific*

ZZZDEBUG!149.tex
Takata, T.|textbf
Ootsubo, T.
Keida, A.
Kasuga, T.
Firusawa, J.
Furusawa, H.
FOOBAR!conference!ADASS 2019
FOOBAR!organisations!ASP
FOOBAR! ascl:1101.010

Moving objects detection in large catalog database with RANSAC

Tadafumi Takata,[1,3] Takafumi Ootsubo,[2] Akira Keida,[1] Toshihiro Kasuga,[1,4] Junko Firusawa,[1] and Hisanori Furusawa[1,3]

[1]*National Astronomical Observatory of Japan, Mitaka, Tokyo, Japan;* tadafumi.takata@nao.ac.jp

[2]*The University of Occupational and Environmental Health, Kita-Kyushu, Fukuoka, Japan*

[3]*The Graduate University for Advanced Studies, SOKENDAI, Mitaka, Tokyo, Japan*

[4]*Nishi-Harima Astronomical Observatory, Center for Astronomy, University of Hyogo, Sayo, Hyogo, Japan*

Abstract. We develop the software for detecting moving objects in large catalog database, which are based on the PDR3-DUD (public data release version 3 for Deep and Ultra Deep survey) of Hyper Suprime-Cam Subaru Strategic Program(HSC-SSP). By comparing the catalogs based on Coadd image and each exposure image, we extract objects without Coadd detection but in exposure image as moving and/or transient object candidates(MTOCs), and then search the line segments(tracklets) which are formed of MTOCs by applying the method based on the RANSAC(RANdom SAmple Consensus), which is extended to 3-dimensional data. This method provides us with faster and numerous detection of moving objects(∼18,000 secure candidates) detection, especially more than 3 times increase in the faint magnitude range.

1. Introduction

Evolution of the solar system from the early to the present stage is one of the important topics in modern astronomy, and we need to know the details about orbits, compositions, and size distributions of small solar system bodies (SSSBs) for the purposes. It is highly required to increase the observational information of small/faint SSSBs to understand the collisional evolution of the solar system. Observations with 8-10 m class large telescopes enable us to investigate the detailed properties on sub-km or smaller SSSBs. SSSBs can be found in wide field imaging data, and detecting fainter ones is the challenging work(Nguyen et al. 2024; Holman et al. 2025). Since SSSBs are moving objects, we have to develop efficient ways to detect new moving objects among numerous astronomical objects. We use photometric data of the Hyper Suprime-Cam Subaru Strategic Program (HSC-SSP) survey(Aihara et al. 2018) for the SSSB detection(Ootsubo et al. 2025). The Hyper Suprime-Cam (HSC) is an optical imaging camera installed at the prime focus of the 8.2 m Subaru Telescope. It covers a wide field of view of 1.8 deg2 with 104 CCDs for science purposes, and thousands of sources are detected in each CCD image in one exposure.

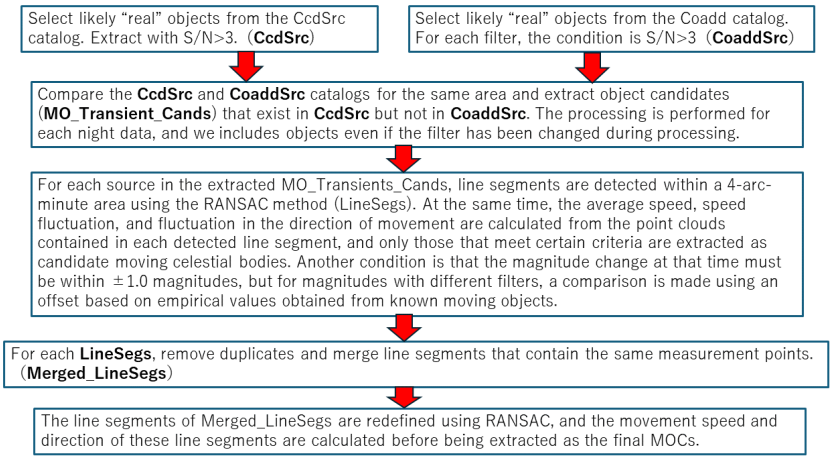


Figure 1. Flow chart of our data processing.

2. Data and Method for detecting Moving Object Candidates (MOCs)

Photometric data of sources detected for all exposure frames obtained with HSC-SSP during Mar. 2014 to Jan. 2020 (public data release 3: PDR3)(Aihara et al. 2022) were obtained through the HSC data analysis pipeline, hscPipe(Bosch et al. 2018). The total number of detected sources across all exposures is 724 millions for PDR3-DUD(deep and ultra-deep survey), and 19 millions in catalog based on the Coadd image data. The measured parameters of all detected sources are stored as tables for each night(160 nights in total) on each exposure and for each tract on the Coadd sources in PostgreSQL server in our system at headquarter of NAOJ in Mitaka and also in Sakura Internet data center. (Takata et al. 2020; Furusawa et al. 2020). We used the RANSAC(Random Sample Consensus) for the detection of moving object candidates, which are thought to form line segments in 3-dimensional space by celestial coordinates(RA,DEC) and time(MJD). RANSAC is an algorithm used to estimate the parameters of a mathematical model from a dataset containing outliers. It works by repeatedly selecting a small, random subset of data points to fit a model, then determining which points best support that model (inliers) versus those that do not (outliers). It is a method that is often used to detect straight lines and other objects in an image. There are several other methods used for detecting moving objects (tracklets) in intra-night wide field large amount of data, such as kd-tree(Kubica et al. 2007) and/or Hough transform(Zhang et al. 2024) and so on. We adopted RANSAC as our method because it is simple and we believe that the analysis results are relatively easy to verify.

3. Application to PDR3-DUD data and its results

In Figure 1, we show the flow chart of the processing for detecting moving objects through comparing exposure based source catalog (CcdSrc) with coadded image based source catalog (CoaddSrc). The basic idea is that objects that are not in Coadd but are in CcdSrc must be limited to moving objects with sufficient velocity or transient objects. By arranging these in two spatial dimensions and one time dimension, a group

of points distributed in a straight line is found, and if the group of points moves at a certain constant speed, it can be recognized as a moving celestial body. In reality, due to the influence of measurement errors, etc., it is necessary to allow for a certain degree of fluctuation in the speed and direction of movement, however it has been thoroughly examined in previous research and it has been determined that there is no problem in considering the movement as uniform linear motion for observation points that are close in time(for example, see the detailed description in Kubica et al. (2007)).

The steps taken this time are as follows: First, for a given night's data, we extracted measurement points that we considered to be celestial objects with S/N > 3 from the catalog based on each exposure image, and then determined whether they existed in the catalog based on the Coadd image, using the criterion of within 0.5 arcseconds in celestial coordinates. Next, for the overnight catalog (MO_Transient_Cands) obtained in the previous processing, each source is placed in the three-dimensional space (RA, DEC, MJD), and we search for and extract sets of four or more linerly distributed sources using the RANSAC (RANdom SAmple Consensus) method extended for three-dimensional point cloud analysis. To search for line segments, an area of 4 arcmin or less is extracted for each measurement point in the RA and DEC directions, and this is used as input. At this point, line segments formed by the same point group are removed to avoid duplication. We used pyRANSAC-3D[1] as a library for our implementation. In the third step, we search for different line segments with the same measurement points, and if found, we merge them to identify the line segment with the highest degree of completeness. In this case, we use the ID number assigned to each measurement point as information for linking. Finally, the newly constructed line segments are merged and the RANSAC method is used again to remove outliers and reconstruct the line information, after which the speed and direction of movement are analyzed to create the final moving object candidates (MOCs) catalog. In Figure 2, we show an example of our results, which is for the data taken in 2019-04-04 by 17 exposures with g-band filter, and 400 moving objects are detected.

4. Summary and Future Prospects

We developed the software for detecting SSSBs in HSC-SSP data (PDR3-DUD) based on the source list produced by comparing sources for each exposure with those in Coadd catalog using RANSAC method, which is well used for detecting line segments in the point cloud. In addition we used information on magnitudes, speed and flags for measurements associated to detected sources by data analysis pipeline. Although we used very simple (or insufficient consideration) method, we were able to obtain satisfying results as our first step to identify numerous faint moving objects in large catalog based on wide and deep imaging data. Our next step is to link the moving objects detected each night, meaning connecting tracklets of the same moving objects to identify the tracks, which is an essential process for creating a moving object catalog. There are already excellent and successful methods established in the world on the linking problem(Kubica et al. 2007; Holman et al. 2018; Nugent et al. 2025), so we plan to use them as a reference to implement something that is suitable for us. We are also planning to extend our activity to much wider survey data such as PDR3-WIDE data of HSC-SSP. We applied our method to a small amount of experimental data and confirmed that it

[1] https://leomariga.github.io/pyRANSAC-3D/

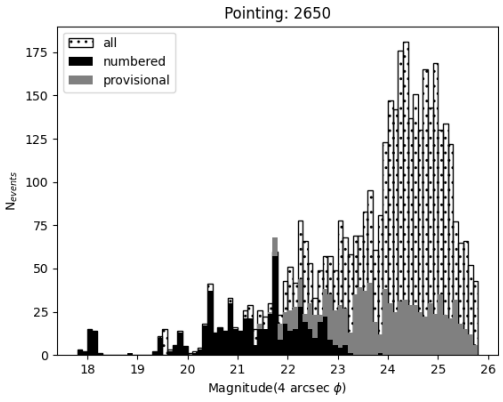


Figure 2. The distribution of magnitudes of the 4-arcsecond aperture for each detection about moving objects. Clearly, new detections(represented by hatched) are more common in faint areas, and in some cases exceed the number of existing detections by five times.

provided sufficient detection performance, although some adjustments were necessary in terms of processing speed and data dependency. We hope to improve processing speed by using the Tsurugi database or an Apache Spark-based system, which we have previously used to detect known moving objects(Takata et al. 2024).

Acknowledgments. This work was supported by JSPS KAKENHI Grant Numbers 23H01217, 23K25913 and 24H00027.

References

Aihara, H., et al. 2018, Publications of the Astronomical Society of Japan, 70, id.S4
— 2022, Publications of the Astronomical Society of Japan, 74, 247
Bosch, J., et al. 2018, Publications of the Astronomical Society of Japan, 70, id.S5
Furusawa, J., Takata, T., et al. 2020, Proceedings of the SPIE, 11452, id. 1145226
Holman, M. J., et al. 2018, Astronomical Journal, 156, 135
— 2025, The Planetary Science Journal, 6, id.152
Kubica, J., et al. 2007, Icarus, 189, 151
Nguyen, T., et al. 2024, Astronomical Journal, 167, id.113
Nugent, C., Tan, N., & Bauer, J. 2025, The Planetary Science Journal, 6, id.18
Ootsubo, T., Takata, T., Furusawa, J., Furusawa, H., Terai, T., Urakawa, S., & Yoshida, F. 2025, in ADASS XXXI, edited by O. Smirnov, & F. others (San Francisco: ASP), vol. 538 of ASP Conf. Ser., 397
Takata, T., Ootsubo, T., Keida, A., Kasuga, T., Furusawa, J., & Furusawa, H. 2024, in ADASS XXXIV (San Francisco: ASP), vol. TBD of ASP Conf. Ser., 999 TBD
Takata, T., et al. 2020, in ADASS XXIV, edited by R. Pizzo, E. Deul, J.-D. Mol, J. de Plaa, H. Verkouter, & R. Williams (San Francisco: ASP), vol. 527 of ASP Conf. Ser., 717
Zhang, B., et al. 2024, Publications of the Astronomical Society of the Pacific, 136, id.054502

Astronomical Data Analysis Software and Systems XXXV
*ASP Conference Series, Vol. **VOL** Compiled: Tuesday 30th June, 2026@ 01:30*
K. Polsterer, D. Bohlender, S. Gwyn, A. Hincks and P. Teuben, eds.
*© **YEAR** Astronomical Society of the Pacific*

ZZZDEBUG!151.tex
ArÃ©valo, M.|textbf
Espinosa, J.
Cook, J. P.
Carasa, Ã.
LeÃ³n, I.
Osinde, J.
PÃ©rez, H. H.
FernÃ¡ndez, M.
MerÃn, B.
Bhatawdekar, R.
FOOBAR!conference!ADASS 2019
FOOBAR!organisations!ASP
FOOBAR! ascl:1101.010

A Multi-mission strategy for long-term data preservation at the ESAC Science Data Centre

María Arévalo, [1] Javier Espinosa,[1] Jonathan Paul Cook, [1] Ángela Carasa,[1] Ignacio León,[2] José Osinde,[1] Héctor Hugo Pérez,[1] Mónica Fernández,[1] Bruno Merín,[3] Rachana Bhatawdekar,[3]

[1]*Starion Spain for ESA, Madrid, Spain;* maria.arevalo@ext.esa.int

[2]*Aurora B.V. for ESA, Madrid, Spain*

[3]*ESA, ESAC, Madrid, Spain*

Abstract. The ESAC Science Data Center (ESDC) develops and operates the Science Archives for ESA missions, providing the scientific community with access to all ESA Planetary, Heliophysics, and Astronomy science data collections. Many of these archives are now approaching the legacy phase, as active missions decline and new ones typically allocate smaller budgets for archiving. ESDC is in fact approaching a turning point where legacy missions will overtake active ones, making it necessary to streamline archive development, operations, and maintenance.

Operating multiple independent archives is resource-intensive, leading to challenges such as software obsolescence, lack of standardization, reduced interoperability, loss of institutional knowledge, and fragmented archive evolution. To address these issues and ensure the long-term preservation and usability of scientific data, we propose the integration of stand-alone mission archives into a unified, scalable backend system: The ESDC Multi-Mission Data Services (EMDS).

The EMDS enables cross-disciplinary, unified data access by implementing standardized interoperability mechanisms, such as the IVOA Table Access Protocol (TAP), and harmonizing interfaces across scientific domains. This strategy enhances scalability, maintainability, and usability, becoming the de facto standard for new missions (e.g., SMILE, Einstein Probe or Proba-3) while supporting the migration of existing archives, ensuring their data is accessible and preserved for long-term scientific use. Additionally, EMDS infrastructure supports the development of common user interfaces, such as the recently released HelioPhysics Archive (HPA) [Cook et al. (2023)]. This article outlines the vision, architecture, challenges, and implementation strategy for evolving ESA's science archives into a single, cohesive, and scalable system.

1. The ESAC Science Data Centre: strategy pillars

At ESAC, the European Space Astronomy Centre, all of the scientific data produced by the majority of ESA's astronomy, planetary science and heliophysics missions are archived and served to the world. The ESAC Science Data Centre (ESDC) [González-Núñez et al. (2017)] is in charge of hosting all these data archives, integrating them under a common framework, and enabling scientists worldwide to access and exploit space science data with increased efficiency and consistency.

ESDC's strategy focuses on three core objectives: **maximizing the scientific exploitation** of ESA's space science archives, ensuring efficient **long-term preservation** of data, software, and knowledge using modern technologies, and enabling **cost-effective archive production** through integration within and across projects. These principles guide the centre's mission to support scientific discovery and maintain the longevity and usability of ESA's valuable data resources.

2. Challenges in long-term data preservation of Science Archives

Preserving science archives over the long-term is increasingly difficult as science archive missions transition from development to operations and eventually into the legacy phase.

The **development phase** focuses on designing and building the archive's foundational system with plenty of resources, prioritizing development, CI/CD, hardware setup, and initial data ingestion, while operational and maintenance tasks play only a minor role.

The **operations phase** shifts the focus to running and maintaining the archive, prioritizing day-to-day operations and continuous data ingestions while development, CI/CD, upgrades, and security tasks continue at lower intensity and hardware or design updates occur only occasionally.

By the time an archive enters the **legacy phase**, resources become very limited forcing focus on minimal maintenance, primarily COTS upgrades and security. Obsolescence, loss of knowledge, and declining interoperability lead to a situation on which archives are no longer actively used, and their maintenance cost remains high, creating an increasingly unsustainable situation over time.

Adopting a **multi-mission architecture** offers a crucial lifeline for legacy archives. By integrating all missions into a shared, scalable infrastructure, ESDC Science Archives require lower long-term costs, achieve improved interoperability, and more reliable access to scientific data.

3. Concept evolution: from Legacy Missions to Multi-Mission

The multi-mission infrastructure has evolved beyond its initial purpose of serving only legacy missions, offering clear benefits for both current and future missions. Integrating missions from the start lowers overall archive costs, avoids legacy-phase issues, and provides a unified, interoperable operational environment that has already become the standard for several new missions. As part of this evolution, existing archives, beginning with the oldest legacy systems, will be migrated to the ESAC Multi-mission Data Services (EMDS).

3.1. Scientific advantages

The transition to a multi-mission infrastructure brings significant scientific benefits.

- **Unified access:** Exploring datasets from multiple missions and domains through a single platform facilitates broader scientific workflows.
- **Cross-mission analysis:** The unified access allows for cross-mission analysis, supporting meaningful data cross-matches wherever they offer scientific value—such

as combining heliophysics observations with planetary measurements or correlating astronomical datasets

- **Domain standards support:** Scientific communities can rely on familiar, interoperable protocols. Astronomy missions benefit from IVOA standards such as TAP, heliophysics datasets support protocols such as HAPI or JPIP, while planetary science archives align with the EPN-TAP standard.

Together, these capabilities create a powerful, scalable scientific ecosystem that maximizes the value of ESA mission data and enhances opportunities for interdisciplinary research.

3.2. Technical advantages

The move toward a unified multi-mission infrastructure provides strong technical benefits.

- **Scalability:** A unified multi-mission infrastructure allows growing organically as new archives are added, whether they are legacy systems, operational missions, or missions still in development, eliminating the need for isolated systems or duplicated environments and supports evolving mission requirements.
- **Maintainability:** Centralizing archives within a shared infrastructure significantly reduces maintenance costs. Compared to managing multiple standalone systems, this approach minimizes duplicated efforts related to updates, monitoring, hardware maintenance, and lifecycle support.
- **Usability and Interoperability:** Enhancing openness and accessibility of scientific data for end users.
- **Flexibility and Client Support:** The infrastructure's robust server-side architecture supports a wide range of client applications. It can reliably serve mission-specific user interfaces, such as the HelioPhysics Archive (HPA[1]), or the Astronomy Multimission Archive (AMA[2]).

4. Multi-mission implementation, challenges and current status

Implementing a multi-mission infrastructure requires a set of core principles designed to ensure cohesion while preserving the flexibility needed for mission-specific requirements.

- **Shared services as a foundation:** Maximizing the use of common services such as authentication, logging, and monitoring, allows all missions to benefit from a robust and consistent baseline.
- **Modular mission-specific extensions:** Mission-specific capabilities are implemented as modular extensions or plugins, enabling customization without fragmenting or duplicating the core platform.

[1] https://hpa.esa.int/hpa
[2] https://ama.esac.esa.int/ama

- **Metadata-driven interoperability:** Rather than enforcing a single unified data model, the approach relies on shared metadata standards to enable interoperability while respecting domain-specific data structures.
- **Coordinated release cycles:** Release planning is coordinated across missions to accommodate differing levels of readiness, with some functionalities introduced through phased deployments before reaching full operational capability.
- **Structured migration strategy:** A well-defined migration path supports legacy archives in transitioning partially or fully into the multi-mission environment.

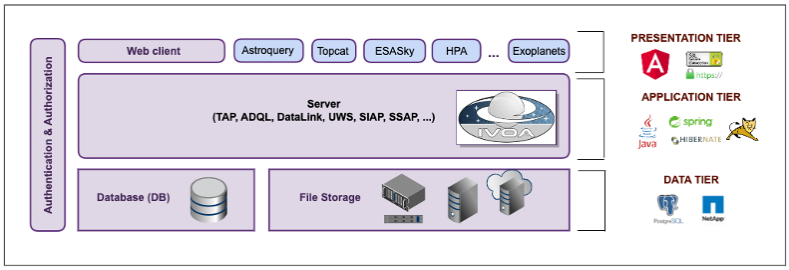


Figure 1. ESDC Multi-mission Data Services high-level architecture and involved technologies.

However, implementing this unified approach introduces collaboration challenges. Achieving consensus across all missions can be difficult, as each mission brings its own scientific priorities, technical legacy, and operational constraints. Furthermore, effective collaboration requires strong organizational coordination across diverse scientific domains, including the alignment of updates, resource planning, and development cycle schedules.

The first public version of (EMDS[3]) was made public in April 2025, offering the access to metadata and data through ADQL in a TAP based server, including data download an login capabilities.

References

Cook, J., Perez, H. H., Carasa, A., Andres, A., Trejo, A., Cruz, M., Masson, A., Middleton, H., Merin, B., & Bhatawdekar, R. 2023, in Data, Analysis and Software in Heliophysics (DASH) 2023, 67

González-Núñez, J., Ríos, C., Salgado, J., Macfarlane, A., Barbarisi, I., Arviset, C., Álvarez, R., Geiger, B., Rourke, L. O., Baldwin, E., Merin, B., Aguado-Agelet, F., & O'Mullane, W. 2017, in Astronomical Data Analysis Software and Systems XXV, edited by N. P. F. Lorente, K. Shortridge, & R. Wayth, vol. 512 of Astronomical Society of the Pacific Conference Series, 141

[3]https://emds.esac.esa.int/service

Astronomical Data Analysis Software and Systems XXXV
*ASP Conference Series, Vol. **VOL** Compiled: Tuesday 30th June, 2026@ 01:30*
K. Polsterer, D. Bohlender, S. Gwyn, A. Hincks and P. Teuben, eds.
*© **YEAR** Astronomical Society of the Pacific*

ZZZDEBUG!155.tex
Heßling, H.|textbf
conference!ADASS 2018
organisations!ASP

Multi-messaging: How can uniform access to astronomical archives be achieved?

Hermann Heßling

[1]*German Center for Astrophysics, Görlitz, Germany;*
hermann.hessling@dzastro.de

Abstract. By combining data from different messengers (electromagnetic radiation, gravitational waves, neutrinos, cosmic rays, ...), one can gain a better understanding of the physics in the universe. A milestone was the merger of a binary neutron star (2017) seen in gravitational waves by LIGO/Virgo, followed by gamma-ray burst, optical/infrared kilonova, X-ray, and radio counterparts.

There are important challenges, for example many messengers are hard to detect, transients often evolve quickly, i.e. a rapid observation of follow-ups is needed, the variety, the velocity, and the volume of data is large in general.

The German Center for Astrophysics (DZA) will provide archives with data from observatories and telescopes around the world. One of the main tasks will be to make it easier for different communities (EM, GW, EM, neutrino, cosmic ray) to cross-match data. This requires standardized data formats, open data, and uniform access, as well as suitable federated data infrastructures

After an overview of the topic, challenges and possible strategies for facilitating access are presented.. The outcome of the discussions with the participants of the Birds of a Feather session (BoF 5) is summarized.

1. Introductory thoughts

The DZA supports the consortium PUNCH4NFDI[1], which is made up of institutions from the fields of astronomy, astroparticle physics, high–energy physics, and nuclear physics. A key objective of PUNCH4NFDI is to contribute the development of a National Research Data Infrastructure (NFDI)[2]. Cloud-based services will be offered, the rough layer architecture of which is shown in Fig. 1. Users can create a container and run it in a remote data center (see the green arrows in Fig. 1) that provides access to storage space (where large volumes of data are archived) and computing hardware.

The DZA aims to support multi–messaging. The cloud architecture shown in Fig. 1 is therefore to be expanded in several respects:

- Astronomical archives offer individual interfaces for accessing their data, i.e., these interfaces should be modified so that access to archives is standardized, thereby making it easier to work with them.

[1]https://www.punch4nfdi.de
[2]https://www.nfdi.de/?lang=en

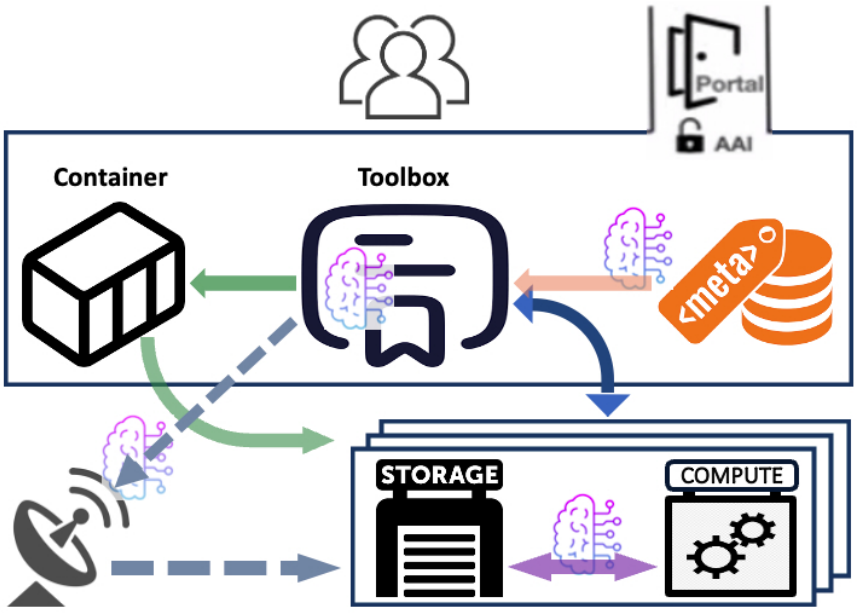


Figure 1. Layered architecture of a federated cloud infrastructure. Illustration by the author.

- A federated AAI (Authentication and Authorization Infrastructure) should be provided to facilitate the management of scientists worldwide who perform analyses on the archive data (see the portal logo in Fig. 1).
- Access to computing power and storage space should be regulated by an accounting system.
- Physicists and astronomers often develop their codes interactively using tools like Jupyter notebook. Therefore, the user should be able to start a container in a remote data center and obtain an interactive connection (see the two–sided blue arrow in Fig. 1).
- Metadata is relevant when data needs to be specified for analysis, and it becomes even more important when it comes to implementing multi–messaging. It is expected that the amount of metadata will grow significantly in the future. As a result, machine learning methods should be available to support the selection of suitable metadata (see the orange arrow in Fig. 1 and the logo with the “neural network brain").
- Most data is stored on tape, and a certain portion is transferred to hard drives, where it can be used for analysis. However, it takes some time to move data from tape to disc. It is therefore advantageous to transfer data in advance, before it is actually needed (Zimmermann et al. 2018). Suitable machine learning algorithms are to be developed for this purpose (see in Fig. 1 and in Fig. 2 the two–sided purple arrow and the logo with the “neural network brain" above it).
- New data from observatories will constantly expand the archives (see the dashed blue lines in Fig. 1). When selecting the data to be archived, it will become

increasingly important to develop machine learning methods (see the logo with the “neural network brain" above the telescope logo).

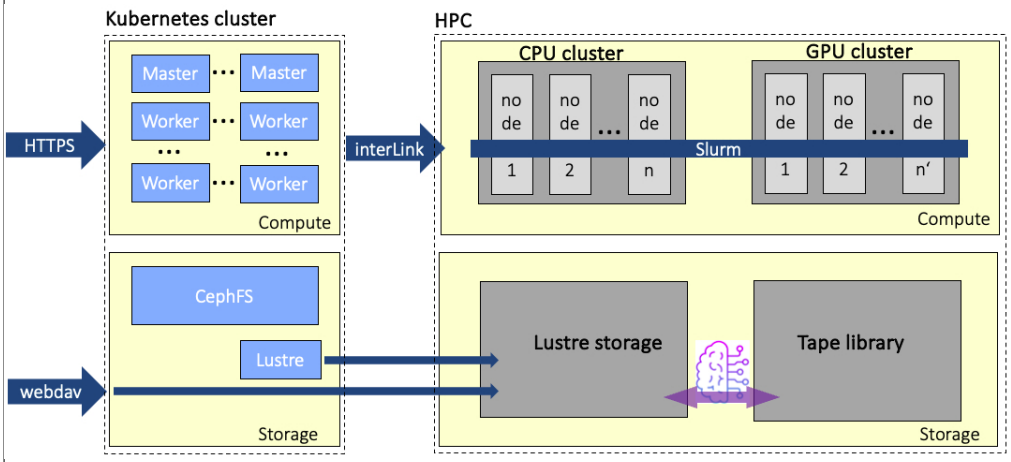


Figure 2. Overview of the architecture of the DZA’s computing hardware. Access to the CPU and GPU cluster (temporarily located in the High–Performance Computing (HPC) center of the Dresden University of Technology during the first years) is provided via the Slurm job scheduling system. Data is stored on tapes or Lustre storage. In addition, a Kubernetes cluster is set up where containers can be started and forwarded to the HPC system using a tool such as interLink. For details, see the poster presentation “Hybrid Cloud & HPC Infrastructure @ DZA" (Borra et al. 2025). Illustration by the author.

2. Important aspects, selected challenges

The Zwicky Transient Facility (ZTF) produces 1.4 TB (compressed) of image data per night and has generated a total of approximately 7 PB of data over the past six years. More than one million exposures have been taken and a total sky area of about 47 million square degrees has been covered, which is roughly 1.5−2 times the total sky coverage expected from the Rubin Observatory LSST over 10 years. ZTF also produces up to one million transient alerts per night, and over 800 million alerts (about 64 TB of data) have been published so far.

Access to data is often facilitated through Jupyter notebooks, which provide a user–friendly entry point for new users. However, resources are finite, even at large institutions, and the diversity of data products – catalogs, images, spectra, and data cubes – raises practical challenges. Decisions regarding the limitation of data downloads appear to be necessary. Storing data in the cloud and accessing it efficiently is costly, and it is unclear whether switching cloud providers will even be possible in the long term, especially from an economic perspective. Additional complexity arises from the need for robust composition, orchestration, and logging of workflows.

Looking ahead, there is a clear need to better support wide-area statistical analyses and large-scale, bulk data processing for machine learning, cross-matching, and recommendation searches. At the same time, the current landscape is fragmented, with

proprietary authentication systems and limited ability to share data or computing resources across platforms.

3. Summary of discussion

There was a lively discussion among the participants of this BoF session. The most important proposals and aspects to be considered in the implementation of this BoF are summarized below.

Implementing multi–messaging for federated archives is a challenging task, especially when it comes to realtime reactions to online alerts. It will be necessary to use suitable AAI solutions to enable a standard access to different archives. The AAI architecture developed by the EOSC Association could be of interest here. Given limited computing and storage resources, access must be controlled by means of accounting.

Due to the strong increase in data volumes in the future, processing should take place close to the data (code–to–the–data paradigm). Containerized workflows simplify analysis on remote data and ensure portability. CANFAR, presented at the ADASS conference, could be a suitable platform.

When developing a uniform interface for accessing different archives, taking the user–perspective into account is most important. User-driven development is widely used in software development, meaning that software development involves end users in every stage of the development process. It is hoped that the complexity of accessing archives and ensuring uniform access can be concealed by suitable middleware. Appropriate support/hotline should be provided for users with different technical backgrounds.

It was pointed out that many individual solutions for accessing specific archives were presented at the ADASS conference, which do roughly the same thing, albeit in different ways. The idea came up to bring these solutions together in order to take a first step toward standardizing access to archives. VO (Virtual Observatory) and TAP (Transient Astrophysics Probe) provide widely used standards that ensure that astronomical data from different telescopes, surveys, and archives can work together.

A working group should be set up to further develop standards, submit proposals for their implementation, and collect open questions.

Acknowledgments. We would like to thank Matthew J. Graham for his support in preparing and moderating the BoF, and Elsa Buchholz for her notes documenting the contributions during the discussion.

References

Borra, S., Buchholz, E., Canbolat, U., Desing Raja, C., Drobek, M., & Haupt, L. 2025, in ADASS XXXV, edited by K. Polsterer, S. Wagner, & et al. (Görlitz: ASP), vol. TBD of ASP Conf. Ser., TBD

Zimmermann, N., Wochinger, T., Fuchs, H., Menzel, M., Nagel, D., Thom, F., Strutz, M., & Hessling, H. 2018, Deep learning for predicting the popularity of datasets. `https://indico.egi.eu/event/3973/contributions/9333/`

Astronomical Data Analysis Software and Systems XXXV
*ASP Conference Series, Vol. **VOL** Compiled: Tuesday 30th June, 2026@ 01:30*
K. Polsterer, D. Bohlender, S. Gwyn, A. Hincks and P. Teuben, eds.
*© **YEAR** Astronomical Society of the Pacific*

ZZZDEBUG!157.tex
Doser, B.|textbf
Polsterer, K. L.
Trujillo-Gomez, S.

Representation Learning for Gaia XP DR3

Bernd Doser, Kai L. Polsterer, and Sebastian Trujillo-Gomez

Heidelberg Institute for Theoretical Studies, Heidelberg, Baden-Württemberg, Germany; bernd.doser@h-its.org

Abstract. We present a novel representation learning framework for the Gaia XP DR3 dataset that leverages two advanced data exploration tools: Spherinator and HiPSter. Spherinator provides a method for learning compact representations of high-dimensional data, including images, point clouds, data cubes, time series, and spectra. Our training process uses variational autoencoders with hyper-spherical latent spaces to efficiently and robustly extract physically meaningful parameterizations of data properties. Our approach explicitly incorporates uncertainties from experiments into representation learning, which produces more robust and physically consistent latent representations. HiPSter generates and serves HiPS-based representations of learned features, enabling the scalable visualization and exploration of the latent space using Aladin-Lite. We demonstrate the scientific potential of our method using the largest spectral dataset available from Gaia XP DR3 and showcase the effectiveness of cross-disciplinary tools developed under the EU SPACE initiative to enhance data-driven astronomy.

1. Introduction

Machine learning provides an effective approach to reduce data to a lower number of dimensions while preserving the similarity of objects in a compressed representation. As part of the EU-funded project "Scalable Parallel Astrophysical Codes for Exascale" (SPACE), we developed the tools Spherinator and HiPSter (Polsterer et al. 2024). Spherinator utilizes a variational autoencoder to reduce high dimensional data including images or spectra to a 2-dimensional spherical space. HiPSter generates Hierarchical Progressive Survey (HiPS) images, which allows for visualization of the latent space using Aladin. Together, Spherinator and HiPSter are essential components of a machine-learning workflow that covers all stages, from data collection and preprocessing to training, prediction, and final deployment (Doser et al. 2025).

This contribution presents the results obtained from the Gaia XP spectra. The Gaia Data Release 3 (DR3) (De Angeli, F. et al. 2023) represents the largest spectroscopic survey ever conducted, encompassing approximately 220 million low-resolution spectra corresponding mostly to stars in the Milky Way.

The Gaia XP spectra have many scientific uses, including determining the atmospheric properties of stars, mapping the distribution of interstellar dust in the Milky Way, and identifying rare and exotic objects. Our goal is to test the capabilities of our tools for exploration and knowledge discovery using the compact representations learned from extremely large datasets.

This survey captures data across two photometric channels: the blue photometer (BP, 330-680 nm) and the red photometer (RP, 630-1050 nm). Together, these are

referred to as XP spectra. Each spectrum is a time-averaged mean spectrum, parameterized using Hermite polynomial basis functions with 55 coefficient amplitudes per channel. This approach allows for efficient storage and transmission of spectral information.

2. The Training

The Preprocessing Engine for Spherinator Training (PEST) is a comprehensive data preparation tool that handles diverse input formats. The continuous data was calibrated using GaiaXPy from 336 to 1021 nm in 2 nm increments (343 data points) and stored in an Apache Parquet[1] format for optimal performance and portability.

The Spherinator employs a variational autoencoder (VAE) operating in a (hyper-)spherical latent space and is trained with PyTorch Lightning. This architecture provides exceptional flexibility and can accommodate diverse data modalities, including images, spectra, data cubes, graphs, and point clouds, through its modular encoder-decoder architecture. For the Gaia spectroscopic data, feature extraction was performed using a one-dimensional convolutional neural network (1D-CNN) with eight layers. This network compresses the initial 343 spectral data points into a 128-dimensional feature map. The intermediate representation is then passed through a single fully connected layer to reduce it to a final 3-dimensional spherical bottleneck.

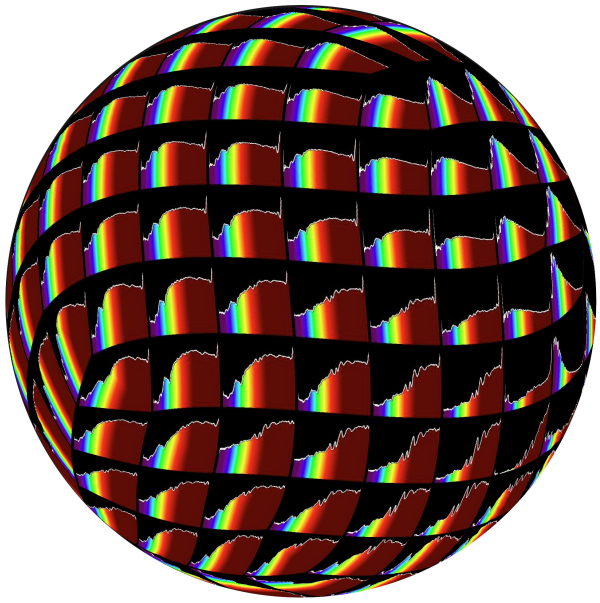


Figure 1. The spherical latent space learned from the Gaia XP DR3 dataset using Spherinator and visualized using HiPSter and Aladin-Lite. Each tile shows a spectrum representing all the similar objects in that region of the latent space. The colors show the simulated visual appearance of each spectrum.

[1] https://parquet.apache.org/

The VAE loss function combines the Kullback-Leibler (KL) divergence with a reconstruction term, balancing regularization of the latent space and data fidelity. The balancing factor is set to $\beta = 0.1$. To account for the observational uncertainties present in Gaia flux measurements, we use a negative log-likelihood (NLL) approach. In this method, the reconstruction loss is calculated assuming that the data is represented by a normal distribution defined by the observed flux and its 1-σ uncertainty.

3. Conclusions

The Spherinator model demonstrates excellent reconstruction capabilities on the very large Gaia XP spectra dataset, successfully capturing the relevant broad spectral features. The uncertainty-aware training approach results in reconstructions that appropriately reflect the confidence levels in different spectral regions, with higher fidelity in well-constrained wavelength ranges and appropriate uncertainty propagation in noisier regions.

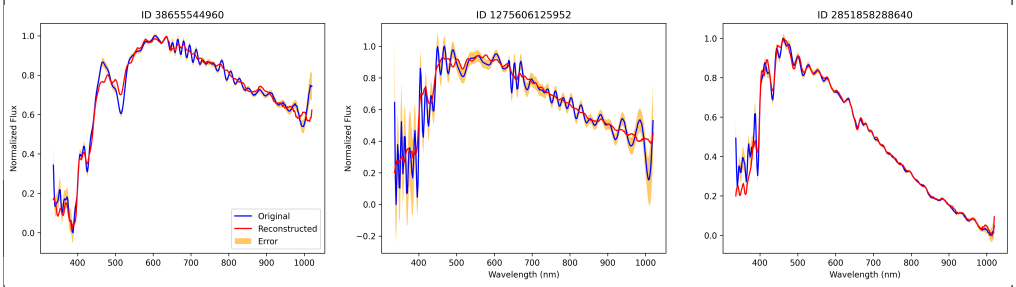


Figure 2. Three examples of Gaia XP spectra showing the original data (blue) and the reconstructions obtained from the learned compressed representations (red). The flux uncertainty is indicated in yellow, and the data has been min-max normalized.

Acknowledgments. We gratefully acknowledge the generous and invaluable support of the Klaus Tschira Foundation. This work has received funding from the European High Performance Computing Joint Undertaking (JU) and Belgium, Czech Republic, France, Germany, Greece, Italy, Norway, and Spain under grant agreement No101093441. Views and opinions expressed are however those of the author(s) only and do not necessarily reflect those of the European Union or the European High Performance Computing Joint Undertaking (JU) and Belgium, Czech Republic, France, Germany, Greece, Italy, Norway, and Spain. Code is available at `https://github.com/HITS-AIN/Spherinator`.

References

De Angeli, F., Weiler, M., Montegriffo, P., Evans, D. W., Riello, M., Andrae, R., Carrasco, J. M., Busso, G., Burgess, P. W., Cacciari, C., Davidson, M., Harrison, D. L., Hodgkin, S. T., Jordi, C., Osborne, P. J., Pancino, E., Altavilla, G., Barstow, M. A., Bailer-Jones, C. A. L., Bellazzini, M., Brown, A. G. A., Castellani, M., Cowell, S., Delchambre, L., De Luise, F., Diener, C., Fabricius, C., Fouesneau, M., Frémat, Y., Gilmore, G., Giuffrida, G., Hambly, N. C., Hidalgo, S., Holland, G., Kostrzewa-Rutkowska, Z., van Leeuwen, F., Lobel, A., Marinoni, S., Miller, N., Pagani, C., Palaversa, L., Piersimoni,

A. M., Pulone, L., Ragaini, S., Rainer, M., Richards, P. J., Rixon, G. T., Ruz-Mieres, D., Sanna, N., Sarro, L. M., Rowell, N., Sordo, R., Walton, N. A., & Yoldas, A. 2023, A&A, 674, A2. URL https://doi.org/10.1051/0004-6361/202243680

Doser, B., Polsterer, K. L., Fehlner, A., & Trujillo-Gomez, S. 2025, Machine Learning Workflow for Morphological Classification of Galaxies. eprint: 2505.04676, URL https://arxiv.org/abs/2505.04676

Polsterer, K. L., Doser, B., Fehlner, A., & Trujillo-Gomez, S. 2024, Spherinator and HiPSter: Representation Learning for Unbiased Knowledge Discovery from Simulations. eprint: 2406.03810, URL https://arxiv.org/abs/2406.03810

Astronomical Data Analysis Software and Systems XXXV
*ASP Conference Series, Vol. **VOL** Compiled: Tuesday 30th June, 2026@ 01:30*
K. Polsterer, D. Bohlender, S. Gwyn, A. Hincks and P. Teuben, eds.
*© **YEAR** Astronomical Society of the Pacific*

ZZZDEBUG!160.tex
Bell, G. S.|textbf
Berry, D. S.
Currie, M. J.
Graves, S. F.
packages!Starlink
packages!ORAC-DR
applications!GAIA
observatories!ground-based
Asian Observatory
observatories!ground-based
Starlink! ascl:1110.012
ORAC-DR! ascl:1310.001
GAIA! ascl:1403.024

Starlink: the 2025A release

Graham S. Bell,[1] David S. Berry,[1] Malcolm J. Currie,[2] and Sarah F. Graves[1]

[1]*East Asian Observatory, 660 N. A'ohōkū Place, Hilo, Hawaii, 96720, USA;* g.bell@eaobservatory.org

[2]*RAL Space, Didcot, Oxfordshire, OX11 0QX, UK*

Abstract. Starlink is an open-source collection of software for astronomy containing tools for data reduction, analysis and visualization. It is currently maintained by the East Asian Observatory to support processing of data from the James Clerk Maxwell Telescope. Recent development has focused on handling problematic data, bug fixes and improving the "summit" pipelines used by the telescope operators.

1. Introduction

With its origins in the UK-based Starlink Project, the Starlink software collection is currently maintained by the East Asian Observatory, primarily to support data from the James Clerk Maxwell Telescope (JCMT). The software is open source and available from GitHub[1] with release packages available from the Starlink website.[2] Starlink releases also include StarJava applications (such as TOPCAT, Taylor 2005) and the ORAC-DR pipeline (Jenness & Economou 2015).

2. The 2025A and 2023A Releases

The 2025A and 2023A releases include the last four years of developments since Starlink was last presented at ADASS XXXI (Bell et al. 2024). While much of the work of preparing the releases went into bug fixes and updates for modern compilers, we present some of the more notable updates in the following sections.

2.1. Developments for Modern Systems

Starlink is now available via Flatpak[3] for easy installation on a variety of Linux distributions. Flatpak is an application packaging and sandboxing system where software is built for a particular standardized platform. The system installs the platform and application and can then run them in an isolated environment. Since the intention of Flatpak is to distribute desktop applications, a single command to run is required — it was de-

[1]https://github.com/Starlink/
[2]https://starlink.eao.hawaii.edu/
[3]https://flatpak.org/

cided to simply make this a shell (running in a terminal) with the Starlink environment set up. An optional extension containing StarJava is also available.

Various scripts were converted from `csh` to `sh` to allow usage on systems without `csh` installed. Python scripts were changed to specify the interpreter as `python3` instead of the unversioned `python` command as this is no longer present on all systems.

Multi-threading has been added to more applications: the GaussClumps method of the CUPID clump finding package and the BLOCK image smoothing application of the KAPPA package. There were various bug fixes related to multi-threading in the NDF and ARY libraries.

2.2. Improved Support for MOC and HiPS

Support for IVOA MOC (Multi-Order Coverage) maps has been improved in the AST library allowing small features to be added to lower resolution MOCs. The outline tracing routine has been made more reliable with the addition of a backtracking algorithm, allowing GAIA (image / data cube viewer) to plot more complex MOCs. An example is shown in Figure 1. GAIA also now only attempts to draws the relevant part of a MOC by first computing the intersection with the area of the current plot. Finally a new command MOCGEN was added to the KAPPA package to create a MOC of the region covered by an image.

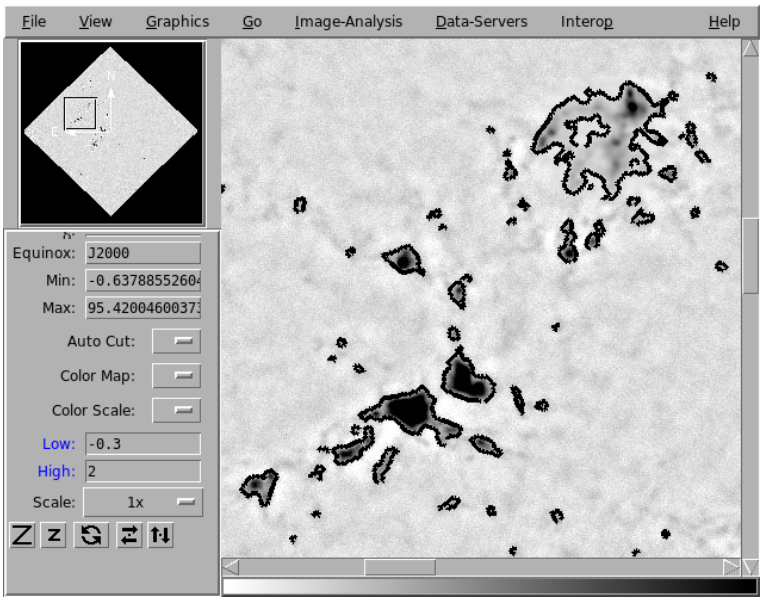


Figure 1. The GAIA image viewer showing a SCUBA-2 map with a MOC of detected emission outlined in black.

The PICARD MAKE_HIPS recipe for constructing HiPS (Hierarchical Progressive Survey, Fernique et al. 2015) structures has been made more efficient and can now take "JSA tiles" (tiles with a HEALPix projection as used in the SCUBA-2 legacy release) as input.

2.3. Updates to JCMT Data Processing

The SCUBA-2 (Holland et al. 2013) and heterodyne pipelines have been updated to produce preview images for pointing, focus and noise observations. While this functionality is available to anyone running the pipelines, it is most useful in the context of the observatory's WORF (WWW Observing Remotely Facility, Jenness et al. 1997) system, which now collects images from the real-time pipelines and makes them available immediately to observatory staff and members of related observing projects. Example preview images are given in Figures 2 and 3.

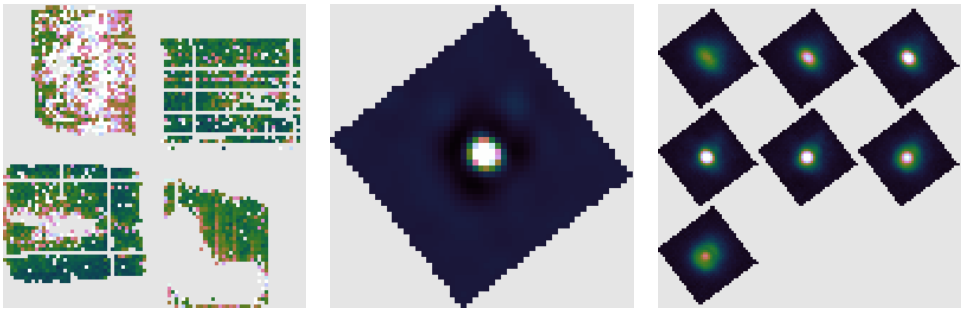


Figure 2. Preview images created by the ORAC-DR "summit" pipeline for SCUBA-2. *Left:* a noise map. *Center:* a small region extracted around a pointing source. *Right:* thumbnails generated by pasting together the planes of a focus cube where each was observed at a different position of the secondary mirror.



Figure 3. Preview images created by the ORAC-DR "summit" pipeline for heterodyne instruments. *Left:* a representative spectrum from a pointing observation. *Center:* a map showing the 5-point pattern used for pointing observations. *Right:* a graph showing the result of adjusting the position of the secondary mirror (SMU) in a focus observation.

A new routine was added to validate the telescope position by comparing the actual and demand positions found in the `JCMTSTATE` extension. This is used by the heterodyne reduction application MAKECUBE allowing the exclusion of individual samples which are out of position.

A number of new recipes were added to the Wesley (Bell et al. 2025) pre-processing pipeline. This software has proven to be a successful way of handling problematic raw data. The new recipes are listed below:

CLEAR_HEADER_SIMULATE Clears the `SIMULATE` header, for use with data known to have been incorrectly flagged as observed with telescope systems in simulation mode.

COPY_BLANK_HEADERS Attempts to fill blank headers using values from other files of the same observation.

FILTER_DOME_OPEN Selects only files where the headers indicate that the telescope dome was fully open. This is useful when the telescope operator was forced to close the dome due to worsening weather conditions but was not able to abort an observation in progress.

FIX_HEADER_IFFREQ Rewrites the `IFFREQ` header if it was written with insufficient precision based on the OCS (Observatory Control System) configuration XML included in the raw data files.

FIX_HEADER_LST Corrects the `LSTSTART` and `LSTEND` headers using the times stored in the `JCMTSTATE` extension of the raw data files.

FIX_HEADER_STEPTIME Sets the `STEPTIME` header from the OCS configuration XML included in the raw data files.

FIX_INCONSISTENT_OBJECT Makes the `OBJECT` headers of all files in a group the same. This is to avoid issues when the data reduction software merges the headers of the input data files. Headers that differ unexpectedly are removed, but the `OBJECT` header is required when archiving the group products.

REMOVE_NAN_VALUES Removes floating-point 'Not a Number' (NaN) values from raw data files, replacing them with the appropriate Starlink bad values.

3. Conclusion

The 2025A release brings the Starlink software collection to a variety of modern systems, including via Flatpak on Linux. It also includes useful updates for JCMT data processing and handling of MOC maps.

References

Bell, G. S., Berry, D. S., Graves, S. F., Currie, M. J., & Draper, P. W. 2024, in ADASS XXXI, edited by B. V. Hugo, R. Van Rooyen, & O. M. Smirnov, vol. 535 of ASP Conf. Ser., 455

Bell, G. S., Graves, S. F., & Parsons, H. A. L. 2025, in ASP Conf. Ser., edited by S. Gaudet, D. Bohlender, S. Gwyn, A. Hincks, & P. Teuben, vol. 538 of ASP Conf. Ser., 377

Fernique, P., et al. 2015, A&A, 578, A114

Holland, W. S., et al. 2013, MNRAS, 430, 2513

Jenness, T., & Economou, F. 2015, Astronomy and Computing, 9, 40

Jenness, T., Economou, F., & Tilanus, R. P. J. 1997, in ADASS VI, edited by G. Hunt, & H. Payne, vol. 125 of ASP Conf. Ser., 401

Taylor, M. B. 2005, in ADASS XIV, edited by P. Shopbell, M. Britton, & R. Ebert, vol. 347 of ASP Conf. Ser., 29

Astronomical Data Analysis Software and Systems XXXV
*ASP Conference Series, Vol. **VOL** Compiled: Tuesday 30th June, 2026@ 01:30*
K. Polsterer, D. Bohlender, S. Gwyn, A. Hincks and P. Teuben, eds.
*© **YEAR** Astronomical Society of the Pacific*

ZZZDEBUG!162.tex
Stoehr, F.|textbf
Schweder, F.
Polsterer, K. L.

U(A)I: User-Interfaces and User-Experience in a World Moving towards AI Agents - and even AI Astronomers

Felix Stoehr[1], Fenja Schweder[2,3], and Kai Polsterer[3]

[1]*European Southern Observatory (ESO), Garching, Germany;* fstoehr[at]eso[dot]org

[2]*University Bremen, MHZ, Bremen, 28359, Germany*

[3]*HITS, Heidelberg, 69118, Germany*

Abstract. The advances of generative AI are staggering and progress is expected to continue at high speed. In the near future astronomers will likely be able to use AI agents to accompany them in the entire process from proposal preparation to archival search, data analysis and publication. How to leverage the advantages for astronomy? How to mitigate the risks? In the first part of this BoF we tried to look into the consequences of this evolution. Moreover, already in May 2025 - and thus just 2.5 years after the first release of ChatGPT - a generative AI has demonstrated fully autonomous PhD-level research capability in computer science (https://www.intology.ai/blog/zochi-acl). It is conceivable, and maybe even likely, that in some years from now AI will be powerful enough to do autonomous research in astronomy. We also tried to anticipate the consequences of this scenario in second part of this BoF.

1. Part 1: Near-term future and AI agents

1.1. Introduction

To introduce the discussion, the BoF participants were first asked to fill out a survey (see Section 1.2) and then were given a short introductory presentation. The presentation covered the current state of agentic AI from the general need for astronomy AI agents perceived by users as revealed in a recent workshop, over AI enhanced software development environments[1] to the Model Context Protocol (MCP), a standardized interface for stateful, conversational mult-step interactions with AI systems. The current evolution is geared towards building an eco system of highly specialized AI agents that are then orchestrated to work together through an additional layer of AI. Two groups have presented at ADASS their own work towards such an agentic AI for astronomy (Serhii et al. (0000), Lorenz et al. (0000)). Commercial agents exist, too[2]. The questions leading into the discussion ask about what the evolution means for astronomers, for observatories, for the tools and interfaces provided, how to leverage advantages and mitigate risks and how observatories should become AI ready.

[1]e.g. https://cursor.com

[2]e.g. scispace.com, resea.ai, cambercloud.com, langchain.com, crewai.com

1.2. Survey

In addition to asking for questions that should be discussed, the survey quantitatively shows the strong involvement of astronomers and observatories already with AI technologies and agentic AI.

Table 1. Survey among BoF participants. 58 participants filled out the survey.

Are you (or scientists around you) using AI to help ...	participants	%
Search/summarize	42	72 %
code	36	62 %
generate images for presentations	16	28 %
write papers	11	19 %
as a end-to-end scientific companion	4	7 %
full autonomous research	0	0 %
Does your institute/observatory ...	**participants**	**%**
plan to provide an AI chatbot to help use your tools and services	25	43 %
have an AI agent written up	13	22 %
plan to have their tools and services become AI ready	13	22 %
plan to provide a Science Platform with integrated AI agent	12	21 %
plan to provide an MCP server	4	7 %

1.3. Discussion

The participants reported about their use of AI tools and agents. Several conclusions emerged from the discussion. The first observation generally agreed upon is that in order to use the AI tools available today and to be sure that they produce correct output, for example writing software, the user has to be knowledgeable in writing software themselves. AI is then just an additional tool and does not replace the software engineer or scientist for coding but speeds their software development up. Ultimately, the full responsibility for the delivered software or science is always the developer or scientist - but they also do get the credit in return.

The participants also noted that while standard tasks and answers were often correct, for more elaborate or niche topics AI returned entirely wrong results, although it has to be noted that just because an AI, e.g. an LLM, has limitations today does not mean that future AI will have the limitations, too. There is also the risk to use the wrong tool, e.g. an LLM instead of a specialized tool for mathematics. There is certainly the danger that a published value is wrong and the the next generation of AI trains on those publications which then reinforces and thus establishes the wrong fact. It is very important to teach students critical thinking and clearly marking AI derived content as such.

In terms of user-interfaces, especially programmatic but also web interfaces, the participants had the view that these interfaces will also remain relevant in a world that moves towards AI agents. In the same way graphical interfaces have added to but not replaced, e.g. a terminal, also the AI agents will become additional tools. And these agents will make use of the infrastructure, e.g. Virtual Observatory (VO) services.

Two advanced projects carried out inside the ADASS community, both presented in Görlitz: Serhii et al. (0000) showed an effort in Canadian Astronomy Data Center (CADC) and Lorenz et al. (0000) introduced work done for the *XMM-Newton* Telescope at the European Space Agency (ESA). Insights that will be gained by those projects will be valuable for the community going forward. These systems rely on a set of pre-defined and well tested agents which then are orchestrated by a higher-level LMM.

The importance of documentation is increasing as it not will be only needed for humans but also for AI agents to be able to use the provided tools properly.

2. A small experiment

In a small experiment, the participants were asked to draw an abstract and phantastic image of a solar system. The drawings were then compared to a drawing generated by AI with the conclusion that while AI is great at recognising patterns people are great at creating patterns.

3. Part 2: Longer-term future and AI astronomers

3.1. Introduction

In the second part we look at the longer term future in which - potentially - an AI system can do astronomy autonomously. At this moment in time it is not clear when (if?) such a situation will arise. There are a number of arguments speaking against of such a development happening very soon. These arguments are based on observations that scaling up the current Large Language models (LLMs) will not provide superintelligence given their lack of really understanding the world. This is shared by a large number of researchers[3]. There are also serious analyses indicating that an AI bubble has built up (e.g. like the dot-com bubble). On the other side, there are also developments that seem to indicate that superintelligence may be arriving sooner than expected. The first fully AI-generated PhD-level publication was accepted to the top Computer Science converence in Natural Langage Processing[4]. In Mathematics, AlphaProof and AlphaGemetry demonstrated to be on par with the best humans at the International Mathematics Olympiade[5]. The world-wide spending on AI research is also staggering at of the order of 9x and 80x larger than the average inflation-corrected yearly spending of the Space Race and the Manhattan Project, respectively. Finally, our analysis shows that the predictions for the reach of superinteligence have been reducing with time from 2055 in the year 2000 to even below 2030 in 2025. Although it is not clear when superinteligence really will be available, the time-scales that are discussed, even by the sceptics, are comparable to time-scales of the planning of major new developments in astronomy[6].

[3] https://aaai.org/wp-content/uploads/2025/03/AAAI-2025-PresPanel-Report-Digital-3.7.25.pdf

[4] https://www.intology.ai/blog/zochi-acl

[5] https://arxiv.org/abs/2502.03544

[6] next.eso.org

3.2. Discussion

Similar to the dot-com bubble, there was a large hype which did not immediately provide the expected results, leading to the crash. However, in the loger run, the predictions then still turned out to be largely true.

The participants reinforced also in this discussion, that user-interfaces also in an AI world will remain highly relevant and that claims about what AI agents will do have to be taken with a grain of salt when it is the entities who want to raise capital making those claims especially as currently the money earned is very small compared to the money spent.

In case AI would be indeed better than humans doing astronomy, then while humans in principle could do astronomy as a hobby, in reality they would not have an income to do so, which would be a huge and very fundamental social crisis. Parts of such an evolution can already be seen, e.g. graphics designers have seen their jobs disappear and the field has been changing so that the lower quality provided by AI has generally been accepted.

The current situation is very turbulent when it comes to the use of AI for example in software development companies, where on the one hand some software developers get fired because they relied too much on AI and can not explain their code any more, but on the other hand some people are hired because they are very good in using software generating tools and are thus very productive although they do not have software development skills as such.

According to the participants, what should be done now in addition to address the benefits and risks as societies as a whole and to rethink how we organize specialization, is to write documentation for LLMs, to use them for tasks like writing draft answers for the helpdesk and to develop APIs that can be used by AI agents later on.

The participants also expressed that they would like to see posters and presentations as well as a continued discussion about this topic at the next ADASS conferences.

Acknowledgments. The authors would like to thank all the BoF participants for the very lively and insightful discussion.

References

Lorenz, E., Aitor, I., & Richard, S. 0000, in Astronomical Data Analysis Software an Systems XXXV (ADASS XXXV), edited by XXX, vol. XXX of Astronomical Society of the Pacific Conference Series, XXX

Serhii, Z., Adrian, D., Hossen, T., & Mrunal, M. 0000, in Astronomical Data Analysis Software an Systems XXXV (ADASS XXXV), edited by XXX, vol. XXX of Astronomical Society of the Pacific Conference Series, XXX

Astronomical Data Analysis Software and Systems XXXV
*ASP Conference Series, Vol. **VOL** Compiled: Tuesday 30th June, 2026@ 01:30*
K. Polsterer, D. Bohlender, S. Gwyn, A. Hincks and P. Teuben, eds.
*© **YEAR** Astronomical Society of the Pacific*

ZZZDEBUG!168.tex
Kruk, S.|textbf
Navarro, V.
Maksym, S.
Gómez, P.
Belanger, G.
Reerink, J.
Arviset, C.
Ferruit, P.
Pettorino, V.
Altieri, B.
de Bruijne, J.
Marinic, F.
López-Caniego, M.
Cánovas, H.
Coia, D.
data analysis!science platforms
surveys!Euclid

Addressing Big Data Challenges for ESA Space Science Missions

Sandor Kruk[1], Vicente Navarro[1], Sebastian Maksym[1], Pablo Gómez[1], Guillaume Belanger[1], Jan Reerink[1], Christophe Arviset[1], Pierre Ferruit[1], Valeria Pettorino[2], Bruno Altieri[1], Jos de Bruijne[2], Filip Marinic[1], Marcos López-Caniego[3], Héctor Cánovas[4] and Daniela Coia[4]

[1]*European Space Astronomy Centre (ESAC), European Space Agency (ESA), Villanueva de la Cañada, Madrid, Spain;* sandor.kruk@esa.int

[2]*European Space Agency (ESA), European Space Research and Technology Centre (ESTEC), Keplerlaan 1, 2201 AZ Noordwijk, The Netherlands*

[3]*Aurora Technology for ESA, European Space Astronomy Centre (ESAC), Villanueva de la Cañada, Madrid, Spain*

[4]*Telespazio UK for ESA, European Space Astronomy Centre (ESAC), Villanueva de la Cañada, Madrid, Spain*

Abstract. The rapid growth of data volumes from the European Space Agency's (ESA) space science missions has created a pressing need for scalable, interoperable, and collaborative analysis environments. ESA Datalabs is a science platform designed to provide researchers with ready-to-use computational environments located close to the ESA Science Archives, enabling data analysis without massive data downloads. The platform integrates Jupyter-based services, GUI tools, shared workspaces, and direct access to ESA multi-mission data.

In particular, the Euclid mission introduces unprecedented data volume and complexity. To support the mission's scientific community, ESA has developed the Euclid Data Space (EDS), a unified portal combining the data discovery (Euclid Archive), visualization (ESASky), and analysis (ESA Datalabs). We present the motivation, architecture, and operational experience of ESA Datalabs and EDS, describe key functionalities, and highlight use cases including the Euclid Quick Data Release (Q1), large-scale cutout generation and machine-learning use cases. We conclude with lessons learned and prospects for Euclid DR1 and other ESA missions.

1. Introduction

The era of Big Data in astronomy is characterised by exponential growth in data volume and complexity. Space science missions, ranging from *HST* and *XMM-Newton* to *Gaia*, *JWST* and *Euclid*, collect and process data products at unprecedented scales. Figure 1 shows the growth of this data stored in the ESAC Science Data Center. *Euclid* alone is expected to generate almost 30 PB of calibrated and intermediate data over its operational lifetime.

Traditional workflows relying on data downloads are no longer feasible at petabyte scale. Also, as scientific workflows evolve toward machine learning, image level analy-

ses, and multi-mission cross-correlation, users increasingly require compute resources close to the data. Science platforms provide a paradigm shift, enabling researchers to analyse mission data directly where it resides.

ESA Datalabs[1] (Figure 2) is the science platform of ESA's Science Directorate (Navarro et al. 2024). It is designed to increase data accessibility, operational efficiency, and scientific return for ESA missions by providing integrated tools, compute resources, and collaborative features.

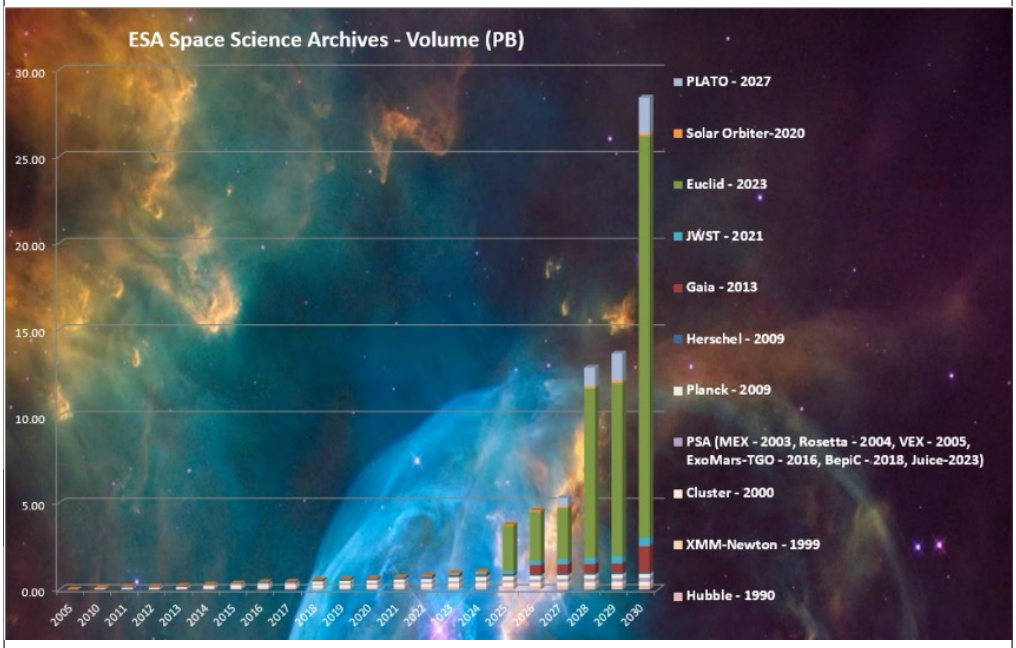


Figure 1. Growth of the data in the ESA Space Science Archives over time, categorised by ESA mission. The ESA *Euclid* mission contributes most of the data in the archives.

2. The ESA Datalabs science platform

Why a science platform? Historically, scientific workflows follow a model in which astronomers search for the data online, download the data needed, then analyse the data locally and publish the results. With petabyte-scale missions, this model becomes inefficient or impossible. Science platforms reverse this paradigm by moving computation to the data.

ESA Datalabs is a science platform that provides researchers with preconfigured scientific environments hosted at ESAC. It is built around three core objectives:

1. **Boost Access to Tools** – Pre-configured, ready-to-use JupyterLab and GUI applications,

[1] https://datalabs.esa.int/

2. **Boost Access to Data** – Direct access to ESA mission archives via NFS-mounted volumes and VO-compliant APIs,

3. **Boost Science Collaboration** – Shared workspaces, environments, and collaborative analysis.

The *bring code to the data* model minimizes the data movement and supports reproducible and collaborative research.

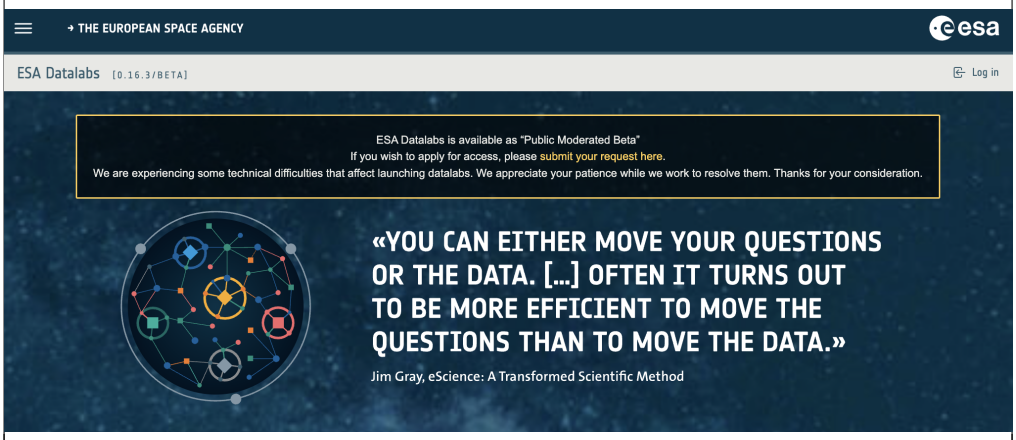


Figure 2. The landing page of `https://datalabs.esa.int/`.

2.1. System infrastructure

The ESA Datalabs computing cluster includes both mission-dedicated and general purposed nodes hosted in the data center at ESAC. Each compute node provides large-memory, multi-CPU, and GPU-enabled resources:

Resource	Specification
System Infrastructure	General pool + mission-dedicated nodes
Memory per node	~ 1.25 TB RAM
CPU per node	~ 128 vCPUs
GPUs per node	2×NVIDIA L4 GPUs
Network	100 Gbps

Table 1. Representative ESA Datalabs Node Resources

The pool of general-purpose nodes supports most missions, while dedicated *Euclid* nodes ensure capacity for several hundred concurrent users during major data releases. ESA Datalabs is built on modern container-orchestration technologies to ensure scalability and consistent user environments. The platform deploys applications as Docker containers managed by Kubernetes, while Rancher provides a unified interface for cluster administration and workload scheduling. This architecture allows the system to scale efficiently during peak activity and ensures reproducible software environments across all users.

2.2. User access and community growth

Access to ESA Datalabs is currently moderated to ensure responsible use of the shared computational resources. Scientists may request access using an institutional e-mail address and by providing either a short statement of motivation or an invitation code provided by ESA missions, collaboration consortia, or by workshop organisers.

Since its launch, ESA Datalabs has experienced rapid growth in its user community. Figure 3 shows the cumulative number of registered users, which has now reached over 3,000. The number of active users (defined as those who have launched at least one application within the preceding 72 hours) is currently approximately 100.

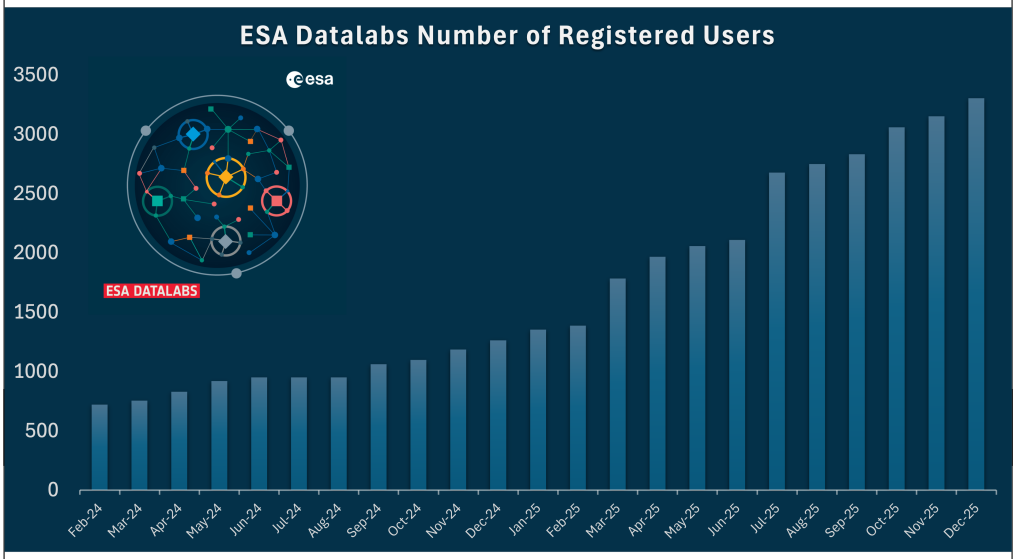


Figure 3. The growth in the number of registered ESA Datalabs users over time.

Science platforms such as ESA Datalabs have proven particularly effective for workshops, hackathons, and hands-on trainings. Because all tools and mission data are provided through a browser-based interface, participants are not required to install software locally or configure computing environments across different operating systems. Organisers can instead prepare shared workspaces, datasets, and instructional material in advance, ensuring a uniform access for all participants and enabling collaborative workflows during hands-on sessions.

To date, ESA Datalabs has supported more than twenty workshops across astronomy, planetary science, heliophysics, and data science disciplines. Recent events include:

- Machine Learning in Heliophysics Workshop[2],
- Working with Euclid Q1 Data[3],
- ESA Datalabs Ariel Hackathon[4],

[2] https://ml-helio.github.io/
[3] https://www.cosmos.esa.int/web/working-with-euclid-q1/home
[4] https://www.ariel-datachallenge.space/esa-datalabs25/

- `.Astronomy`[5].

These activities have contributed substantially to community adoption of platforms by lowering technical barriers and promoting best practices in reproducible, cloud-enabled research.

3. Euclid Data Space (EDS)

The ability to enable large data releases directly within science platforms depends on seamless integration between three traditionally distinct components: the long-term data archives, the tools used for data visualisation, and the interactive environments used for scientific analysis. Historically, these systems evolved independently, reflecting the different needs of their respective times. Science archives were established first, providing curation, metadata management, and preservation of mission data. Later, tools such as ESASky[6] offered essential visualisation capabilities for exploring complex, multi-instrument datasets stored in the archives. More recently, science platforms such as ESA Datalabs have emerged to support interactive analysis, multi-mission access, and collaborative development directly next to the data.

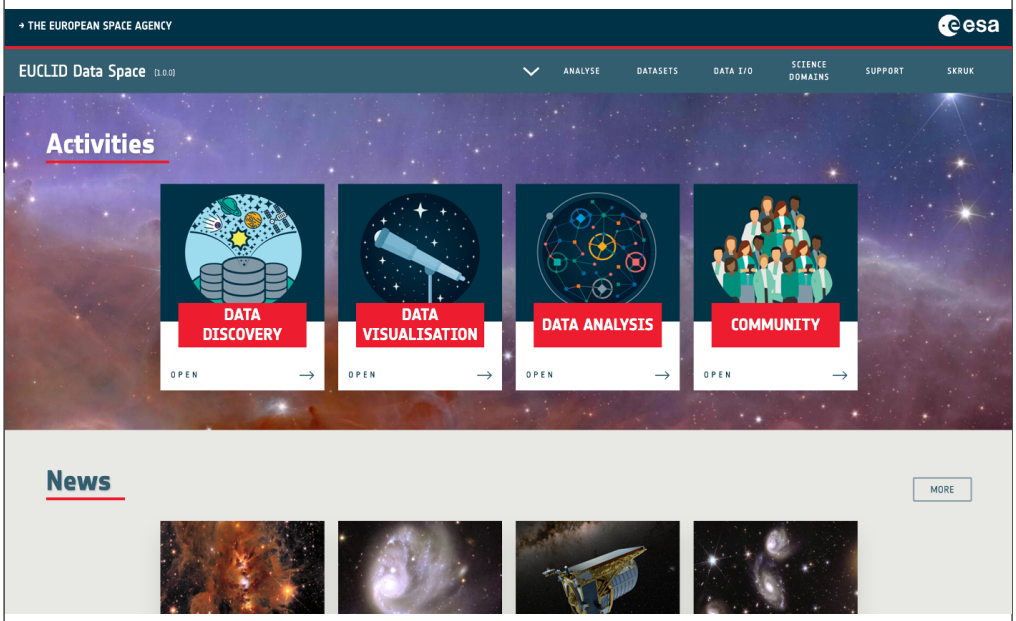


Figure 4. Representative layout of the Euclid Data Space portal (`https://euclid.dataspace.esa.int/`), showing the integration of the Euclid Archive, ESASky visualisation services, and ESA Datalabs analysis environments.

The *Euclid* mission brings these requirements together. To support both discovery and analysis workflows, Euclid Data Space[7] (EDS) provides a unified, single-entry

[5] `https://www.dotastronomy.com/`
[6] `https://sky.esa.int/`
[7] `https://euclid.dataspace.esa.int/`

portal that integrates the Euclid Archive, ESASky visualisation services, and ESA Datalabs computational environments (Figure 4). Through this integration, users can move easily from querying the archive, to visual inspection, to interactive analysis without transferring large datasets locally or navigating multiple websites and interfaces.

EDS implements role-based access control, enabling differentiated view for public users and members of the Euclid Consortium. It further provides topic-oriented views, embedded applications, and mission-tailored interfaces that simplify navigation across *Euclid*'s diverse data products. An internal release of EDS for the Euclid Consortium took place in late 2025, with the first public release planned for 2026.

The use of ESA Datalabs and EDS to enable the *Euclid* use cases and data releases is detailed in the following subsections.

3.1. Euclid pilot studies

In preparation for *Euclid*'s early data releases, ESA Datalabs supported a series of pilot studies to validate data-access strategies, show representative analysis workflows, and evaluate performance for large-scale in-situ processing. These studies covered a broad set of science and operations use cases, including: identifying moving Solar System objects across repeated Euclid exposures; detecting transient sources such as supernovae; reprocessing individual observations with alternative sky-subtraction strategies to enhance low-surface-brightness structures; stacking Euclid spectra; and using the Cosmology Likelihood for Observables in Euclid (CLOE; Euclid Collaboration et al. 2025b) for cosmology forecasts.

The pilot studies provided important input for the platform optimisation, helping to assess resource needs, refine user workflows, and identify missing functionalities prior to the wider release. They also established proven patterns for scalable analysis within ESA Datalabs, thereby informing the design and integration of *Euclid*-specific tools within the Euclid Data Space.

3.2. Euclid Quick Data Release (Q1)

The first *Euclid* data release supported on the platform was the Euclid Quick Data Release (Q1)[8], published on 19 March 2025. Q1 consisted of three deep fields covering a total of $\sim$ 63 deg^2 and approximately 35 TB of data (Euclid Collaboration et al. 2025a). ESA Datalabs played a central role in enabling rapid community access to these data products by providing:

- mission-mounted *Euclid* data volumes,
- a dedicated *Euclid Q1* JupyterLab application (see Figure 5),
- astroquery-based programmatic access via VO-compliant protocols, returning file paths to data already mounted within ESA Datalabs,
- a suite of ready-to-use Jupyter tutorials for *Euclid*, including examples for accessing Q1 data via astroquery, generating image cutouts, visualising images and spectra, and performing data downselection using ADQL queries[9],

[8] `https://www.cosmos.esa.int/web/euclid/euclid-q1-data-release`

[9] These notebooks are openly accessible on GitHub at `https://github.com/ESA-Datalabs/Euclid-Q1`

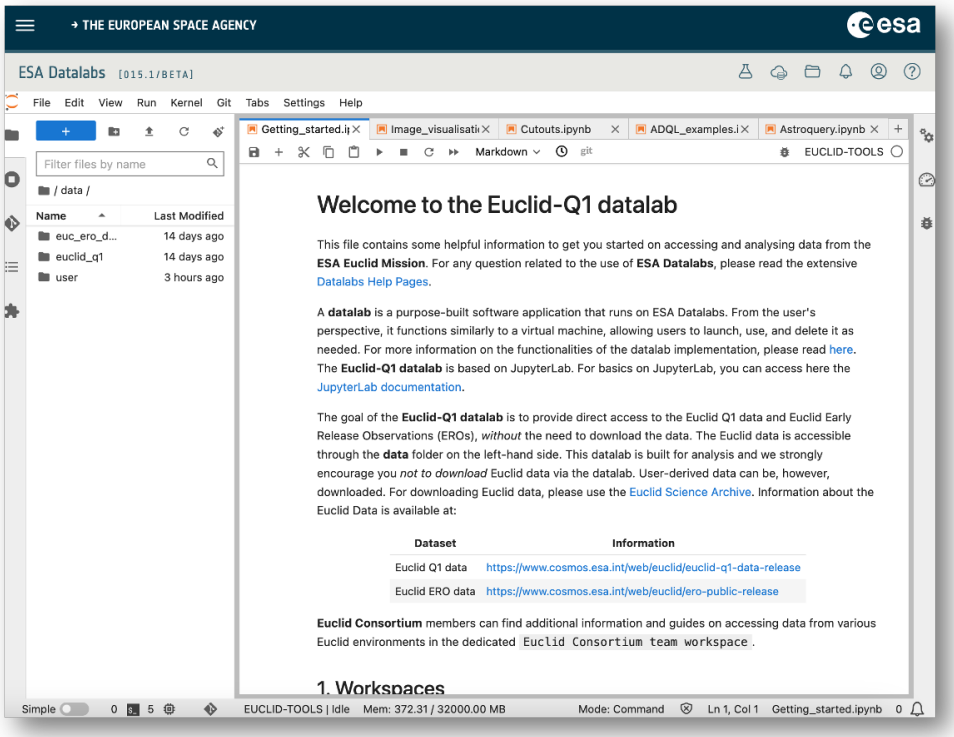


Figure 5. View of the *Euclid Q1* application in ESA Datalabs, showing the mounted `euclid_q1` volume, a *Getting Started Guide* describing the available data products, and links to example Jupyter Notebooks.

- collaborative workspaces for teams,
- monitoring dashboards (e.g. Grafana) to track system load and user activity.

The Q1 release represented the first large-scale use of ESA Datalabs for scientific analysis. 11 out of 27 Euclid Q1 science papers (about 40%) reported using ESA Datalabs for tasks such as catalogue queries and cross-matching, direct inspection of FITS files, large-scale cutout generation, AI-based inference workflows, and collaborative analysis. A survey of participating scientists provided further information into their workflows, highlighting missing software packages, desired improvements to the user experience, and priorities for future development.

3.2.1. Representative use case: Galaxy Zoo for Euclid

One representative example of early science enabled by ESA Datalabs is the creation of a large, high-quality morphological catalogue of *Euclid* galaxies using a combination of volunteer classifications via *Galaxy Zoo* and machine learning (Euclid Collaboration et al. 2025e)[10].

[10] `https://www.esa.int/Science_Exploration/Space_Science/Euclid/Euclid_Galaxy_Zoo_help_us_classify_the_shapes_of_galaxies`



Figure 6. Workflow for the Galaxy Zoo: *Euclid* project. Image cutouts are generated within ESA Datalabs and submitted to Galaxy Zoo for volunteer classification, which is used to train the Zoobot machine-learning model (Walmsley et al. 2023). Zoobot is then run in inference mode on ESA Datalabs to produce large-scale morphological catalogues of *Euclid* galaxies.

The first step involved producing large-scale image cutouts of bright and extended galaxies directly within ESA Datalabs. These cutouts were generated and provided to *Galaxy Zoo* for human classification by almost 10,000 volunteers, supplying the training data required for Zoobot to achieve high performance in automated identification of morphological features such as spiral arms, bars, tidal distortions, and signatures of galaxy mergers. Zoobot was then run in inference mode within ESA Datalabs to predict the morphology of approximately 380,000 galaxies. The current catalogue represents only $\sim$ 0.4% of the galaxies expected to be observed over *Euclid*'s Wide Survey, implying that the final morphological catalogue will exceed previous surveys by at least an order of magnitude.

ESA Datalabs played a key enabling role in this effort by providing scalable in situ cutout production, shared workspaces, and GPU-backed environments for running machine learning pipelines. The same methodology has been applied to other early science cases. Using a combination of AI-based preselection, citizen science inspection, expert vetting, and subsequent modelling, a first catalogue of roughly 500 galaxy–galaxy strong-lens candidates has been built (Euclid Collaboration et al. 2025d), demonstrating the potential of the platform for large-scale discovery workflows[11].

3.3. Preparing for Euclid DR1

Euclid Data Release 1 (DR1) represents a huge increase in data volume and complexity compared to the Q1 release[12]. At approximately 1.5 PB and covering $\sim$ 1900 deg^2 (Euclid Collaboration et al. 2025c), DR1 is more than thirty times larger than Q1, placing fundamentally new demands on storage, data access, and interactive analysis. Unlike Q1, DR1 is far too large to be downloaded and analysed locally by most users, making analysis on science platforms not only advantageous but essential.

[11] https://www.esa.int/Science_Exploration/Space_Science/Euclid/Euclid_opens_data_treasure_trove_offers_glimpse_of_deep_fields
[12] https://www.cosmos.esa.int/web/euclid/data-release-dr1

DR1 was prepared for Euclid Consortium access in late 2025, with a public release planned through the Euclid Data Space in 2026. In anticipation of this milestone, substantial work has been undertaken to ensure that ESA Datalabs, ESASky, and the Euclid Archive operate together as a coherent and robust system capable of supporting thousands of users during the release.

Current preparation efforts focus on scaling the underlying infrastructure and refining the user experience. Key areas of development include:

- scalable data-mounting strategies,
- fair-share computing for hundreds of active users,
- enabling GPU-backed ML workflows,
- EDS seamless integration with Euclid Archives, ESASky, and Datalabs with a single-sign-on.

4. Conclusions

ESA Datalabs has emerged as a central component of ESA's data-analysis ecosystem, providing a unified, scalable, and analysis-ready environment for a wide range of space-science missions. By collocating computation with mission data and offering a consistent, browser-based interface, ESA Datalabs lowers the barriers to accessing and analysing complex, PB-scale datasets. Its integration within the Euclid Data Space (EDS) represents a major step toward fully end-to-end scientific workflows, enabling users to transition smoothly from data discovery, to visualisation, to interactive analysis without downloading large data volumes. The Euclid Q1 data release showed the platform's scientific impact, and preparations for DR1 highlight its essential role for future ESA missions facing PB-scale data.

Beyond *Euclid*, ESA Datalabs is already supporting a diverse set of missions and scientific use cases across astronomy, planetary science, and heliophysics. Workshops, hackathons, and training events have shown that science platforms are efficient mechanisms for building expertise, fostering collaboration, and aligning best practices across disciplines. As the landscape of data-intensive research continues to evolve, platforms such as ESA Datalabs will play an increasingly central role in enabling open, reproducible, and community-driven science.

The experience gained through the development and operation of ESA Datalabs and the Euclid Data Space has highlighted several key lessons:

- Bringing compute to the data is essential for PB-scale missions.
- Science platforms complement rather than replace archives.
- VO protocols ensure data discoverability via the archives and effective use through mounted volumes on the science platforms.
- Platforms must support both novice and expert users.
- Clear communication of resource limits manages expectations.

Acknowledgments. We thank the ESAC Data Science Section, ESAC Science Data Centre, the Programatic IT Section, the ESA Datalabs and Euclid Data Space Development Teams, and our industrial partners (CGI, ACRI-ST, Edisoft, Uninova). We also thank ESA mission scientists and archive teams for their support and feedback.

References

Euclid Collaboration, Aussel, H., Tereno, I., Schirmer, M., Alguero, G., Altieri, B., & Balbinot, E. e. a. 2025a, arXiv e-prints, arXiv:2503.15302. eprint: 2503.15302

Euclid Collaboration, Cardone, V. F., Joudaki, S., Blot, L., Bonici, M., Camera, S., Cañas-Herrera, G., Carrilho, P., Casas, S., Davini, S., Di Domizio, S., & Farrens, S. e. a. 2025b, arXiv e-prints, arXiv:2510.09118. eprint: 2510.09118

Euclid Collaboration, Mellier, Y., Abdurro'uf, Acevedo Barroso, J. A., Achúcarro, A., & Adamek, J. e. a. 2025c, A&A, 697, A1. eprint: 2405.13491

Euclid Collaboration, Walmsley, M., Holloway, P., Lines, N. E. P., & Rojas, K. e. a. 2025d, arXiv e-prints, arXiv:2503.15324. eprint: 2503.15324

Euclid Collaboration, Walmsley, M., Huertas-Company, M., Quilley, L., Masters, K. L., & Kruk, S. e. a. 2025e, arXiv e-prints, arXiv:2503.15310. eprint: 2503.15310

Navarro, V., del Rio, S., Angel Diego, M., Lopez-Caniego, M., Marinic, F., Kruk, S., Reerink, J., & Arviset, C. 2024, in Space Data Management. Studies in Big Data, vol. 141, 1–13

Walmsley, M., Allen, C., Aussel, B., Bowles, M., Gregorowicz, K., Slijepcevic, I., Lintott, C., Scaife, A., Jabłońska, M., Karchev, K., Lanzieri, D., Mohan, D., O'Ryan, D., Saiguhan, B., Suárez, C., Guerra-Varas, N., & Velu, R. 2023, The Journal of Open Source Software, 8, 5312

Astronomical Data Analysis Software and Systems XXXV
*ASP Conference Series, Vol. **VOL** Compiled: Tuesday 30th June, 2026@ 01:30*
K. Polsterer, D. Bohlender, S. Gwyn, A. Hincks and P. Teuben, eds.
*© **YEAR** Astronomical Society of the Pacific*

ZZZDEBUG!171.tex
Mitschang, A.|textbf
Lemson, G.
Taghizadeh-Popp, M.
Thakar, A
software!open source
software!user
interfaces!web-based
computing!resources

Open SciServer: The (Now Open) Collaborative Science Platform

A. Mitschang, G. Lemson, M. Taghizadeh-Popp, and A. Thakar

The Johns Hopkins University Institute for Data-Intensive Engineering and Science , Baltimore, MD, USA; arik@jhu.edu

Abstract. SciServer is a mature, high-impact science platform that unites data, compute, and collaboration for research and education. Originally developed for astronomy, it provides access to large observational (e.g., SDSS, HEASARC) and theoretical (e.g., Millennium, Indra simulations) datasets alongside free computational resources and collaborative features such as groups and file sharing. The platform has since expanded to materials science and engineering, turbulence, oceanography, and precision medicine and genomics, with successful deployments at Brookhaven National Laboratory and the Max Planck Institute for Extraterrestrial Physics.

Recently, an NSF sustainability grant enabled Open SciServer — bringing the platform code and governance to the open source community. The consolidated codebase is now public on GitHub with streamlined build and release processes. Anyone with a Kubernetes cluster can now install SciServer using Helm, either on premises or in public cloud environments. This paper introduces Open SciServer, its scientific impact, underlying architecture, and opportunities for community contribution.

1. Introduction

SciServer is a mature production science platform designed to bring together data, compute, and collaboration for research and education (Taghizadeh-Popp et al. 2020). In practice, the platform emphasizes three pillars - data, collaboration, and compute - and offers a unified user experience via a publicly accessible instance at `https://apps.sciserver.org`. Building on a decade of operational experience, we introduce Open SciServer, the open source evolution of the platform that invites community participation in development, deployment, and governance.

The impact of SciServer is well established in astronomy and beyond. In astronomy, co-locating large datasets with compute and enabling team science through group-based sharing has accelerated discovery and training. Beyond astronomy, SciServer has been adopted in materials science and engineering, turbulence studies, oceanography, precision medicine, and genomics, and it has supported classroom instruction across these domains.

In this paper we provide a platform overview; highlight representative datasets; describe the system architecture and deployment approach; summarize the transition to open source and its governance; and outline opportunities for the astronomy community to adopt and contribute

2. Platform overview

SciServer's data services enable researchers to work directly against hosted data without large-scale network transfers. Through web interfaces, users can submit SQL queries to relational databases and retrieve results for immediate analysis, read and process remote file stores as if they were local, and run workflows via popular interactive development environments such as Jupyter, Rstudio, and VSCode, among others.

Collaboration features are embedded throughout the platform so that science teams can organize and share resources effectively. A flexible resource management system allows administrators and project leads to allocate compute and storage aligned with institutional policies and project goals. Users create and manage groups reflecting research collaborations, course cohorts, or working groups, and apply sharing policies at the level of file volumes and databases. These features support both open-science and controlled-collaboration use cases, translating well to classroom settings where instructors need to scaffold assignments, provide shared datasets, and manage per-student or per-team workspaces.

Compute environments are designed to be close to the data (bringing the analysis to the data rather than vice-versa) and easy to use, lowering the threshold for complex analyses. Researchers can launch interactive environments and execute complex analyses without provisioning their own infrastructure, enabling exploratory science, training, and reproducible workflows. Combined with access control features, varying levels of performance hardware can be made available on an as-needed basis – for example, limited memory and number of CPU cores made generally available to the public, while GPUs are provided to a subset of users for specific research activities, as is the case for SciServer@JHU.

3. Datasets

The astronomy community has leveraged SciServer to integrate observational and theoretical resources in a single, collaborative workspace. SciServer has long been the official science platform for the Sloan Digital Sky Survey (SDSS) catalog data, providing multi-modal access to science-ready datasets for all public SDSS data releases via the Catalog Archive Server (CAS), along with integrated access to the full file-based datasets provided by the Science Archive Server (SAS). More recently we have partnered with NASA HEASARC (High Energy Astrophysics Science Archive Research Center) to mirror their archive within the platform and provide managed science environments pre-loaded with a full suite of mission-specific tools.

SciServer's versatility is evidenced by its use outside astronomy. Users can query slices of large scale turbulence simulations from the Johns Hopkins Turbulence Database (JHTDB; Li et al. (2008)). In materials science, we have partnered with NIST (US National Institute of Standards and Technology) to bring the Additive Manufacturing Benchmark Test Series (AM-Bench) to SciServer. Users working with single-cell genomics data can now bring up the *cellxgene* (Megill et al. 2021) viewer with a single click to analyze data of a variety of catalogs including the cytometric atlas of combination immunotherapy in pancreatic cancer described in Sidiropoulos et al. (2026).

Many of these datasets also serve a role in a classroom setting. Instructors prepare template notebooks with instruction material to go through with the class in a consistent environment, referencing reliable curated datasets. They are able to use the Groups

feature to share class materials and give layered access to students, TAs and teachers. SciServer has been prominently featured in the "SDSS in the Classroom" sessions at AAS meetings.

4. System architecture and deployment

SciServer integrates data access, collaboration and identity, and compute services behind a unified web interface. Users authenticate once and gain access to shared datasets, group-managed resources, and compute environments. The hosted instance at JHU serves as both production environment and reference for deployments. Services are implemented as a collection of interconnecting applications, primarily written using the Spring web framework (`https://spring.io/`) in Java and communicate via JSON wire format. While all components expose HTTP+JSON API endpoints for interacting with the system, a typical end-user will interact via HTML+JavaScript based web graphical interfaces that utilize said API endpoints. Figure 1 shows an example of the architecture of SciServer as deployed on Kubernetes, highlighting the connections between components and user interaction.

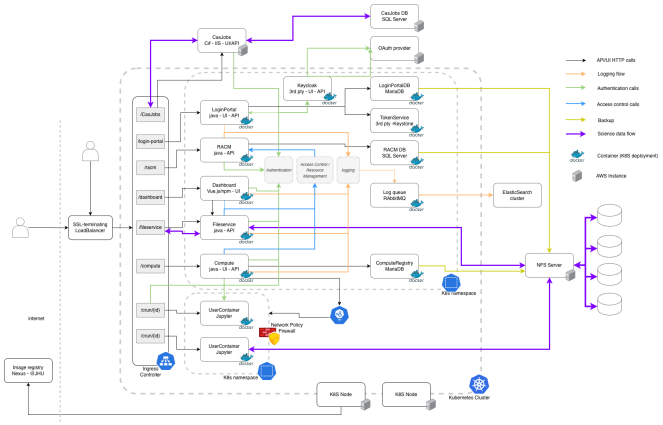


Figure 1. SciServer architecture as deployed on Kubernetes with typical features

For deployment, Open SciServer embraces cloud-native practices. The platform can be installed on any compatible Kubernetes cluster using Helm, enabling on-premises deployment to meet data residency and security requirements, or public cloud deployment for elasticity and managed services. This portability allows data providers and research groups to deploy local instances near sensitive datasets while maintaining consistent user experience across sites.

Partner installations underscore the platform's portability and institutional value. At Brookhaven National Laboratory, SciServer provides streamlined access to GPU resources for computational research. At the Max Planck Institute for Extraterrestrial Physics, a local deployment provides access to eROSITA data, bringing familiar tools to proprietary or restricted datasets and enabling appropriate collaboration.

5. From SciServer to Open SciServer

SciServer originated under the National Science Foundation (NSF) DIBBs (Data Infrastructure Building Blocks) program and is now sustained via an NSF CSSI (Cyberinfrastructure for Sustained Scientific Innovation) grant focused on long-term sustainability through community governance and open development. This shift formalizes the project's commitment to open practices and broadens its contributor and stakeholder base.

As part of the open-source transition, the project consolidated code into a single public repository located at `https://github.com/sciserver/opensciserver`. During the merge, we upgraded Java dependencies, applied style checks and corrected many (though not all) style violations, implemented build automation and release packaging, and improved documentation to lower adoption and contribution barriers. These changes support reliable, repeatable releases and make it easier for external partners to deploy, extend, and maintain the platform.

We are presently working with the Johns Hopkins Open Source Programs Office to develop a governance structure that will facilitate decision-making and maintain an overall roadmap for the project. Though not complete, we welcome community contributions and invite users to deploy the system on their infrastructure. We envision such users, along with our current partner institutions that are operating local instances, to propose fixes and useful new features to the platform. Those with the technical expertise can join the effort by creating pull requests and/or issues for feature requests within our public github repository.

6. Opportunities

SciServer brings together robust data access, integrated collaboration, and accessible compute to advance research and education across astronomy and multiple scientific disciplines. By opening the platform as Open SciServer - with a unified public repository, modernized build and release workflows, and a path to community governance — the project invites broad participation in sustaining and extending its impact. Institutions can now deploy SciServer with Kubernetes and Helm, researchers can contribute features and documentation, and the community can co-create the next generation of data-intensive, collaborative science workflows and class curricula.

Acknowledgments. SciServer core development was originally funded by the NSF (award #ACI-1261715, Data Infrastructure Building Blocks). SciServer features and additions related to open source release was funded by the NSF (award #2311791, Cyberinfrastructure for Sustained Scientific Innovation).

References

Li, Y., et al. 2008, Journal of Turbulence, 9, N31. eprint: 0804.1703

Megill, C., et al. 2021, bioRxiv. eprint: https://www.biorxiv.org/content/early/2021/04/06/2021.04.05.438318.full.pdf, URL `https://www.biorxiv.org/content/early/2021/04/06/2021.04.05.438318`

Sidiropoulos, D. N., et al. 2026, Cancer Immunology Research. Epub ahead of print, URL `https://doi.org/10.1158/2326-6066.CIR-25-1126`

Taghizadeh-Popp, M., et al. 2020, Astronomy and Computing, 33, 100412. eprint: 2001.08619

Astronomical Data Analysis Software and Systems XXXV
*ASP Conference Series, Vol. **VOL** Compiled: Tuesday 30th June, 2026@ 01:30*
K. Polsterer, D. Bohlender, S. Gwyn, A. Hincks and P. Teuben, eds.
*© **YEAR** Astronomical Society of the Pacific*

ZZZDEBUG!176.tex
Michel, L.|textbf
Kleiver, L.
Maggi, P.
computer
languages!Python
software!development!DevOp
astronomy!time domain
pipelines
astronomy!X-ray
FOOBAR! ascl:1101.010

Pipeline of the X-Ray camera of the SVOM mission

Laurent Michel,[1] Louis Kleiver,[1] and Pierre Maggi[1]

[1]*Université de Strasbourg, CNRS, Observatoire Astronomique de Strasbourg, UMR 7550, F-67000 Strasbourg, France;* laurent.michel@astro.unistra.fr

Abstract. The **S**pace-based multi-band astronomical **V**ariable **O**bjects **M**onitor mission, a joint Sino-French collaboration launched in 2024, is designed to detect, localize, and study gamma-ray bursts (GRBs) and other high-energy transients. Among its onboard instruments, the *Microchannel X-ray Telescope* (MXT) plays a central role by providing follow-up X-ray observations of GRB afterglows and other transient phenomena with high sensitivity in the 0.2-10 KeV range. To ensure accurate and timely scientific exploitation of MXT observations, a dedicated ground processing pipeline has been developed. This pipeline automatically ingests raw event lists, performs calibration, background and time filtering, and corrects instrumental effects, and produces science-ready data products such as images and light curves and spectra of detected sources. This poster describes the architecture and key components of the MXT data processing pipeline, highlighting its modular design and integration with the broader SVOM ground segment. We also show results from real datasets that demonstrate the ability of the pipeline to meet mission performance requirements.

1. Introduction

SVOM is equipped with four primary scientific instruments designed to detect, localize, and characterize gamma-ray bursts (GRBs) across a broad range of wavelengths (Gotz 2025). The ECLAIRs (Godet et al. 2014) telescope provides initial detection and localization capability in the X-ray and low-energy gamma-ray domain (4-250 keV). Complementing this, the Microchannel X-ray Telescope (MXT) (Plasse et al. 2025) observes GRBs in the soft X-ray range (0.2-10 keV), enabling detailed follow-up measurements of the early afterglow. To extend coverage into higher energies, the Gamma-Ray Burst Monitor (GRM) measures the spectra of high-energy bursts from 15 to 5000 keV. In the visible domain, the Visible Telescope (VT) records the prompt optical emission and the early optical afterglow immediately following a gamma-ray burst.

The satellite operates in a low Earth orbit with an inclination of 30 degrees, an altitude of 625 km, and an orbital period of approximately 96 minutes, providing frequent revisits and continuous monitoring capabilities. Space-based observations are further enhanced through an extensive network of ground-based facilities. Two Ground Follow-up Telescopes (GFT), located in Mexico and China, deliver rapid and precise measurements of GRB coordinates, essential for multi-wavelength follow-up. Additionally, the Ground-based Wide Angle Camera (GWAC) array in China monitors the visible sky to capture the prompt optical emission of selected bursts from the ground.

2. French Mission Ground Segment

The French Mission Ground Segment (FMGS) is structured to support the complete spectrum of mission operations and scientific activities. Within this framework, the French Science Center (FSC) is responsible for most science-related tasks, including data reception, processing, archiving, and distribution to the scientific community. In parallel, activities directly associated with instrument operations such as calibration, performance monitoring, and the preservation of instrument-specific expertise are conducted by dedicated Instrument Centers (the EIC, MIC, and GIC). Together, the FSC and the Instrument Centers form the French Science Ground Segment (FSGS), operating as an integrated unit. This organizational structure ensures coherent coordination of scientific and technical responsibilities within the French contribution to the mission.

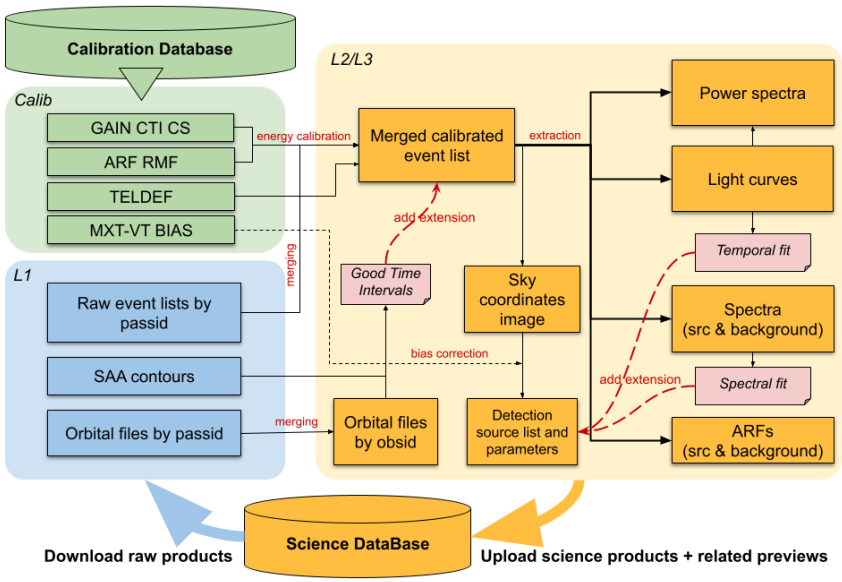
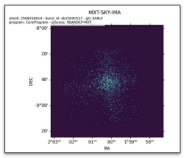


Figure 1. MXT processing workflow.

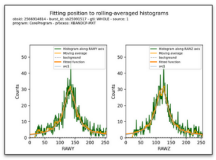
3. The MXT Pipeline

The processing workflow (Fig 1) is composed of approximately 40 individual Python tasks are organized into four sequential stages: data access and preparation, event calibration and filtering, X-ray image generation and source detection, and finally source product extraction. (Fig 3 2). To manage these operations, the FSC has developed Pipeline-Bricks, a generic task scheduler capable of executing any workflow defined in this modular structure. Workflow execution is controlled through a standard REST API shared across all FSC pipelines, providing functions such as initiation, termination, log retrieval, and status monitoring. Each pipeline instance is deployed as a Docker image within the FSC infrastructure, ensuring consistency and portability. Processing is automatically initiated by a global orchestrator whenever a full set of required input files

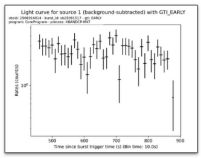
becomes available in the Science Database (SDB), enabling efficient and timely data handling.



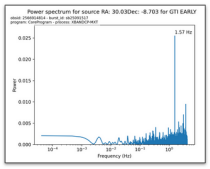
MXT sky image of **sb25091517**, a flare of **1RXS J020013.6-084106** actually. The cross PSF is typical of lobster-eye optics.



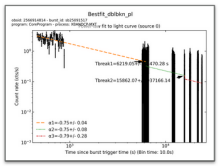
The source position is first computed from the Gaussian fits of the accumulated count histograms along the detector axis.



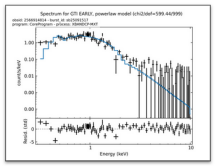
Light curve of the early observation (500" from the event) which is meant to enclose the GRB afterglow with a lower noise.



Power spectrum of the early light curve



Light curve covering the whole observation. The blanks correspond to the unavailability of the instrument (e.g. Earth occultation).



Fitted spectrum of the source. Different models are tested (BB, PL) on both early and whole observation time

Figure 2. Some outputs for event sb25091517 (2025-09-16T00:10:00Z).

4. Development Cycle

All ten participating institutes adhere to a unified workflow for software development and deployment. This process is supported by continuous integration (CI) scripts that perform extensive checks at every stage. Contributors across all institutes follow the same development cycle, managed through a local GitLab instance that oversees both development and deployment activities. Any modification committed to the Git repository triggers the CI system, which verifies code quality and test coverage; the process is automatically stopped if any check fails. Once validated, all FSC modules are packaged and delivered as Docker images that meet established quality and coverage standards. New features are first deployed within an integration infrastructure that mirrors the production environment, ensuring realistic testing conditions. Deployment to the production node occurs only after successful validation and formal approval of the integration results.

5. Conclusions

The mission's design combining a versatile suite of space-based instruments, a coordinated ground-based observational network, and a robust scientific operations infrastructure provides an unprecedented capability for the rapid detection and comprehensive characterization of gamma-ray bursts. The integration of the French Science Center

with dedicated Instrument Centers ensures that both scientific data processing and instrument expertise are maintained at the highest standards. A modular, automated and rigorously validated data-processing workflow further guaranties that scientific products are delivered efficiently and reliably to the community. This is reinforced by a unified development and deployment strategy across all ten participating institutes, allowing consistent software quality and operational stability throughout the mission. Together, these elements form a cohesive and resilient framework that maximizes the scientific return of the mission and ensures its long-term reliability. The collaborative architecture from hardware to software, from orbit to ground positions the mission as a major contributor to the study of high-energy astrophysical phenomena.

Acknowledgments. We would like to thank all FMGS contributors, as well as CNES and all Chinese partners, who made it possible to implement this complex infrastructure.

References

Godet, O., Nasser, G., Atteia, J. ., Cordier, B., Mandrou, P., Barret, D., Triou, H., Pons, R., Amoros, C., Bordon, S., Gevin, O., Gonzalez, F., Götz, D., Gros, A., Houret, B., Lachaud, C., Lacombe, K., Marty, W., Mercier, K., Rambaud, D., Ramon, P., Rouaix, G., Schanne, S., & Waegebaert, V. 2014, in Space Telescopes and Instrumentation 2014: Ultraviolet to Gamma Ray, edited by T. Takahashi, J.-W. A. den Herder, & M. Bautz, vol. 9144 of Society of Photo-Optical Instrumentation Engineers (SPIE) Conference Series, 914424. eprint: 1406.7759

Gotz, D. 2025, in Frontier Research in Astrophysics - IV, 66

Plasse, C., Götz, D., Meuris, A., Moita, M. F., Ferrando, P., Favier, L., & Ceraudo, F. 2025, Nuclear Instruments and Methods in Physics Research A, 1080, 170717. eprint: 2506.02271

Astronomical Data Analysis Software and Systems XXXV
*ASP Conference Series, Vol. **VOL** Compiled: Tuesday 30th June, 2026@ 01:30*
K. Polsterer, D. Bohlender, S. Gwyn, A. Hincks and P. Teuben, eds.
*© **YEAR** Astronomical Society of the Pacific*

ZZZDEBUG!177.tex
Gutiérrez-Sánchez, R.|text
Solano, E.
Falcón-Barroso, J.
Allende-Prieto, C.
Alonso-Asensio, I.
Caon, N.
Marín-Galán, C.
Pérez-Prieto, J. A.
de Vicente, A.
Westendorp, C.
FOOBAR! ascl:1101.010

CIELOS and the Spanish Virtual Observatory: Exposing astronomical data from Canary Islands Observatories through VO Protocols

Raúl Gutiérrez-Sánchez,[1] Enrique Solano,[1] Jesús Falcón-Barroso,[2,3] Carlos Allende Prieto,[2,3] Isaac Alonso Asensio,[2,3] Nicola Caon,[2,3] Carlos Marín Galán,[2,3] Jorge Andrés Pérez-Prieto,[2,3] Ángel de Vicente,[2,3] and Carlos Westendorp Plaza[2,3]

[1]*Centro de Astrobiología (CAB, CSIC-INTA), Villanueva de la Cañanda, Madrid, Spain*

[2]*Instituto de Astrofísica de Canarias, San Cristobal de La Laguna, Santa Cruz de Tenerife, Spain*

[3]*Departamento de Astrofísica, Universidad de La Laguna. San Cristobal de La Laguna, Santa Cruz de Tenerife, Spain*

Abstract. CIELOS (Canary Islands data cEnter for astronomicaL Observations and Simulations) is an initiative led by the Instituto de Astrofísica de Canarias (IAC) to manage, process, and archive the large volumes of observational and simulation data generated by the Observatorios de Canarias (OCAN). To ensure that its data are Findable, Accessible, Interoperable, and Reusable (FAIR), CIELOS will be fully integrated into the Virtual Observatory (VO) ecosystem from the outset. The Spanish Virtual Observatory (SVO, CAB-CSIC/INTA) will play a key role in this process by providing VO-compliant services that expose CIELOS data to the international community following IVOA standards. This poster presents the architecture of this collaborative model and the mechanisms that ensure global accessibility, interoperability, and scientific impact OCAN data within the VO framework.

1. Introduction

The 21st century has brought a fundamental shift in the way astronomical success is measured. Traditionally, the prestige of an observatory depended on the aperture of its telescopes. However, the current bottleneck is no longer data acquisition but the ability to process and exploit information. Following the examples of successful archives such as those at ESO or the Hubble Space Telescope, the IAC has launched CIELOS to ensure that data from the Observatorios de Canarias (OCAN) remain accessible, interoperable and reusable.

CIELOS is not merely a storage solution; it is a comprehensive ecosystem for the acquisition, processing, and dissemination of both observational data and numerical simulations. Its mission is to prevent a technological gap and solidify the role of OCAN as an essential node in the global data network.

2. The CIELOS Technical Infrastructure

The core of the project is a modular and scalable data center hosted by IAC. The infrastructure is built upon three main technical pillars:

2.1. Computing and Storage Resources

CIELOS leverages high-performance resources to handle the massive data volume generated by OCAN.

- Massive Storage: designed to manage petabytes of information.
- HPC and GPUs: systems optimized for intensive analysis and machine learning.
- High-Speed Networks: utilizing the IAC's high-capacity connectivity to ensure efficient data transfer.

2.2. Automatic Processing Pipelines

To minimize human error and reduce processing times, CIELOS implements automatic pipelines adapted to each instrument. These pipelines transform raw data into research-ready products through standardized reduction processes.

2.3. Data Modeling (CAOM)

To ensure internal consistency and global interoperability, CIELOS explores the Common Archive Observation Model (CAOM), developed by the Canadian Astronomy Data Centre (Dowler & Gaudet (2025)). This model allows for robust metadata standardization, facilitating integration with global archival ecosystems (such as Dowler 2019).

3. Federated Data Interaction: Integration with the Spanish Virtual Observatory (SVO)

The SVO is the Spanish national Virtual Observatory initiative, an official component and member of the International Virtual Observatory Alliance (IVOA). As such, it represents Spain within the global framework of the Virtual Observatory and contributes to its governance, standardization, and technical development.

The cornerstone of CIELOS's data discovery strategy is the collaboration with the Spanish Virtual Observatory (SVO, CAB-CSIC/INTA). This partnership adopts a federated access model to balance local storage needs with global visibility.

3.1. Metadata Synchronization (CAOM Harvesting)

To maintain synchronization between the IAC and the SVO, the teams are evaluating the CAOM harvesting mechanism. This allows the SVO to automatically ingest a version of the available metadata from the IAC's local archives.

3.2. Data Interaction Workflow

The diagram in Figure 1 illustrates the interaction between the user, the SVO, and the CIELOS infrastructure. A federated data access model is designed to manage and expose astronomical data from the Canary Islands Observatories (OCAN) through a collaboration between CIELOS and the Spanish Virtual Observatory (SVO).

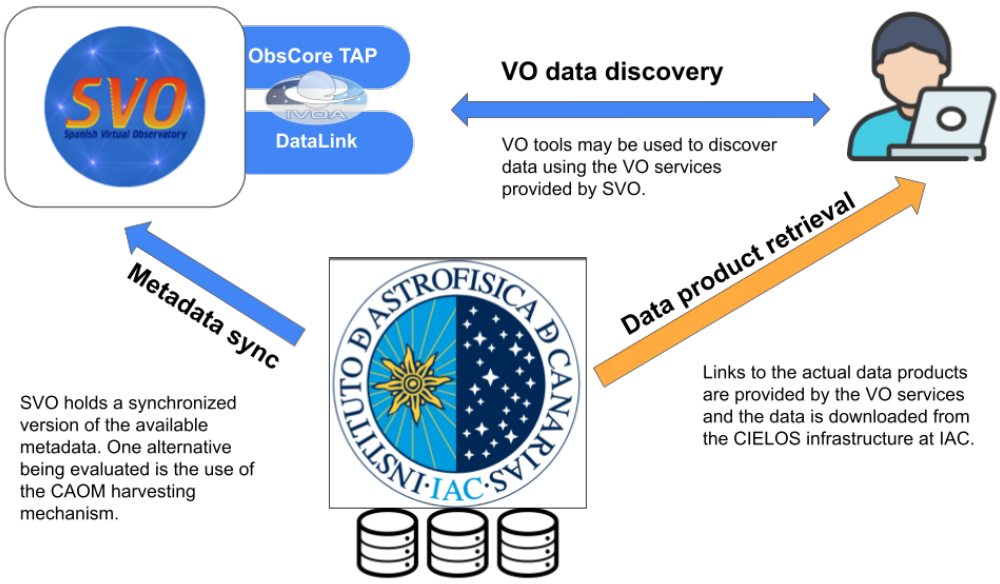


Figure 1. Federated data access workflow.

- Data acquisition and processing at IAC: The process begins at the CIELOS infrastructure, which manages the massive data volume generated by OCAN telescopes. These data are processed using automated pipelines to create high-level scientific products, and their metadata are standardized using the Common Archive Observation Model (CAOM) structure.

- Metadata Synchronization: The SVO and IAC utilize a mechanism known as CAOM harvesting, which allows the SVO to hold a synchronized version of the available metadata while the primary data files remain at the IAC.

- Discovery through SVO: From the user's perspective, the SVO acts as the federated access point for global visibility. Researchers use Virtual Observatory (VO) tools and protocols, such as ObsCore (Louys et al. (2017)) and Table Access Protocol (Dowler et al. (2019)), to search and discover data within the SVO catalogs.

- Data Retrieval through DataLink (Dowler et al. (2023)): Once a user identifies the desired data, the VO services at the SVO provide the necessary connections through the DataLink protocol. This protocol provides links to the actual data products, enabling the user to download the files directly from the CIELOS infrastructure at the IAC.

This model ensures that, while discovery is centralized and standardized at the SVO following IVOA guidelines, the physical storage and delivery of petabyte-scale information remain optimized within the local IAC infrastructure.

4. Conclusions

CIELOS represents a vital evolution for the Observatorios de Canarias, moving from simple observation sites to comprehensive data service providers. By combining massive storage, high-performance computing, and a deep partnership with the Virtual Observatory, the IAC ensures that the scientific legacy of the OCAN remains at the forefront of the international astronomical community for the coming decades.

References

Dowler, P. 2019, in Astronomical Data Analysis Software and Systems XXVI, edited by M. Molinaro, K. Shortridge, & F. Pasian, vol. 521 of Astronomical Society of the Pacific Conference Series, 446

Dowler, P., Bonnarel, F., Laurent, M., Demleitner, M., & Taylor, M. 2019, Table Access Protocol, IVOA Recommendation. Version 1.1, URL https://www.ivoa.net/documents/TAP/20190927/

Dowler, P., Bonnarel, F., Michel, L., Demleitner, M., & Taylor, M. 2023, IVOA DataLink, IVOA Recommendation. Version 1.1, URL https://www.ivoa.net/documents/DataLink/20231215/

Dowler, P., & Gaudet, S. 2025, Common Archive Observation Model, IVOA Working Draft. Version 2.5, URL https://www.ivoa.net/documents/CAOM/20251031/

Louys, M., Tody, D., Dowler, P., Durand, D., Michel, L., Bonnarel, F., Micol, A., & the IVOA DataModel working group 2017, Observation Data Model Core Components and its Implementation in the Table Access Protocol, IVOA Recommendation. Version 1.1, URL https://www.ivoa.net/documents/ObsCore/20170509/

Acknowledgments. Grant PID2023-146210NB-I00 funded by MICIU/AEI/10.13039/501100011033 and by ERDF/EU.

Astronomical Data Analysis Software and Systems XXXV
*ASP Conference Series, Vol. **VOL** Compiled: Tuesday 30th June, 2026@ 01:30*
K. Polsterer, D. Bohlender, S. Gwyn, A. Hincks and P. Teuben, eds.
*© **YEAR** Astronomical Society of the Pacific*

ZZZDEBUG!184.tex
Jeschke, E.|textbf
Kackley, R.
Tanaka, M.
Kim, M.

Porting a Multi-Object Spectrograph Mask Design App from IDL to Python

Eric Jeschke,[1] Russell Kackley,[1] Makoto Tanaka,[1] and Minkyeong Kim

[1]*Subaru Telescope, National Astronomical Observatory of Japan*

Abstract. We present a case-study in migrating a mask design software application for a Multi-Object Spectrograph (MOIRCS, operating at Subaru Telescope) from an IDL (Interactive Data Language) implementation to a pure Python one. The port accomplished several goals, including: (1) freeing users from onerous licensing restrictions, and (2) improving the overall stability and responsiveness of the program, and (3) improving the prospects for future upgrades and maintenance. We explain the purpose and use of the program, describe the practical procedures used in the port, and finally show and describe the user interfaces of both the old and the new versions.

1. Introduction

MOIRCS (Multi-Object Infrared Camera and Spectrograph) provides imaging and low-resolution spectroscopy from 0.9-2.5 microns over a 4x7 arcmin field of view, operating on the Subaru Telescope at the summit of Maunakea, Hawaii. A popular feature of this instrument is the MOS mode, in which a custom-cut mask can be used to obtain spectra for multiple objects within the same field. MOIRCS slit masks consist of a number of slits and alignment holes cut into a thin, flexible mask. Several masks can be loaded into the instrument prior to observation and then inserted into the optical path during observation. The alignment holes are used to line up with stars of sufficient magnitude and an alignment procedure during observation shifts and rotates the focal plane so that stars appear in the holes and the slits are aligned precisely with the objects for which they are configured. After that, a series of exposures can be taken to obtain the necessary spectra.

When the instrument was commissioned in 2006(Ichikawa et al. 2006) an IDL-based program was developed to allow potential users to create the mask design files necessary for the observatory to cut the needed masks prior to observation. The output of the mask design program is a simple, tabular file with coordinates and lengths measured in millimeters for the mask cutting process.

2. How the IDL version works

The IDL program works by loading a FITS image large enough to cover the MOIRCS field of view. The image needs to have a valid world coordinate system and should be rescaled to 0.117 arc seconds per pixel. The program provides a graphical overlay showing the MOIRCS detector outline and overall field of view, as shown in Figure 1 (left). The user then chooses a grism (and optionally can change some of the grism

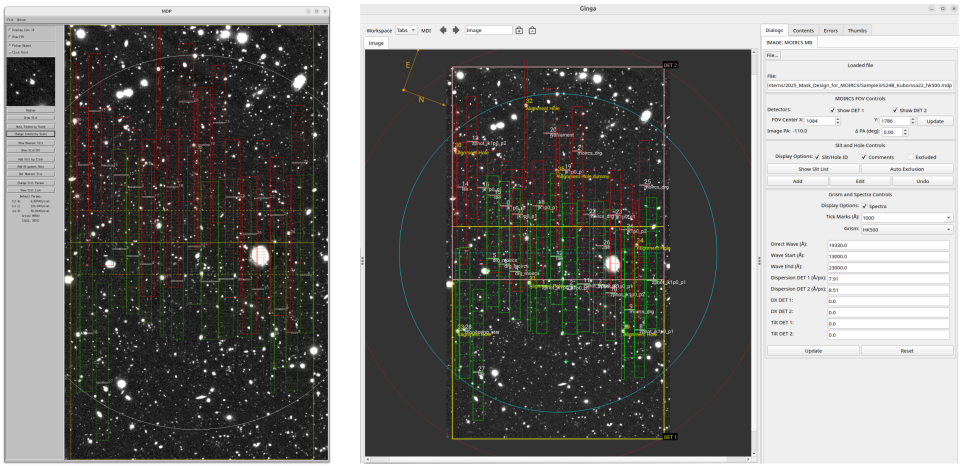


Figure 1. Screenshot showing the IDL version on the left, and the new MOIRCS Mask Builder plugin running in Ginga on the right.

parameters). Following this, the user places slits and holes within the frame and can configure the slit lengths and widths as needed. These objects can be placed by clicking on the image or by measuring the center of an object in the image. This configuration can be saved in a ".mdp" (MOIRCS Design Program) file, which is a simple ASCII CSV-formatted file containing the types, positions and attributes of the slits and holes in pixel units. An MDP file can be loaded to initialize the program if the user needs to make subsequent adjustments.

The spectral dispersion of the slits and holes on the detectors can be visualized with an option, allowing one to see if there are overlaps or other issues that could cause contamination of the spectra. The user adds as many holes and slits as necessary and adjusts their locations and parameters till they are satisfied that the design is reasonable. Finally, the user concludes the process by a final save of the the MDP file and also exporting an output SBR file (with the ".sbr" extension) that contains the physical coordinates and measurements used by the mask cutting machine.

3. Issues with the IDL Version

The IDL version of the mask builder program has several issues that were problematic for our use case:

- IDL itself has licensing restrictions, which makes it difficult to configure which computers can run the software. The license server is inside the observatory firewall, so the user either has to be logged into the IDL host, or run a special "virtual machine" that includes an IDL run-time with the application code and access to the license. However, due to export control policies, a registration and verification process is necessary on the vendor's website(SOFTWARE 2025) to download the virtual machine software. Between the license fee, licensing restrictions and the virtual machine registration issues the whole process is onerous compared to open-source solutions.

- The virtual machine running the IDL program was notoriously unstable and would crash very easily without any clear idea of the cause. Needless to say, this did not impart a sense of confidence in the software.

- Fewer and fewer staff are knowledgeable in IDL application development and the original author of the software has largely moved on to other responsibilities.

4. Porting from IDL to Python

Given the issues described above, we were interested in porting the application from IDL to Python. Python has good support for astronomical applications with good base frameworks like *astropy* or *skyfield*, and a number of GUI frameworks to choose from.

We acertained that the Python version of the application would need to (1) Load and display a FITS file, and adjust contrast or set cut levels on the image for good visibility of the overlays; (2) Draw overlays on the image to show the field of view, detector positions, slit and hole positions with labels, spectral dispersion, and (3) provide a UI with controls for loading and saving MDP files, adjusting parameters for slit position, width, etc. and exporting the mask cutting SBR file.

Subaru Telescope uses the *Ginga* package(Jeschke et al. 2012), an open-source toolkit for building astronomical viewers in Python, in a number of different applications. This viewer already provides many of the basic features needed for this program including good support for viewing FITS files and drawing graphical overlays. During the summer of 2025, our intern Minkyeong Kim took on the task of turning this IDL program into a Ginga plugin within an approximate 7 week time span. It is worth noting that Kim had no knowledge of astronomy or spectroscopy before starting the project, but did have a good understanding of Python programming.

The basic procedure for guiding Minkyeong toward the goal was as follows. First, we explained in general terms what the IDL program should be doing and provided a copy of the IDL code. Second, We gave her a few different example sets of FITS files with input (mask design) and output (mask cutting) files. Third, we gave her time to experiment with the IDL program and understand the basic UI and how that worked. After this initial introduction, we introduced Ginga, showing the features that covered much of the needed functionality and explaining how the plugin architecture works (and showing a few examples of plugins). We guided her to get a skeleton of a new plugin working with a very simple UI and then answered questions as needed as she iterated toward a solution.

After the seven week internship, the MOIRCS Mask Builder plugin was fully functional. Our staff continued to polish and do mild refactoring on the program for an additional few weeks to make the operation a bit more intuitive and maintainable, as well as adding some additional features.

5. Using the MOIRCS Mask Builder Plugin

Using the MOIRCS Mask Builder is fairly straightforward coming from the old workflow. After starting the plugin, all files can be loaded from the plugin UI, shown in the right side of Figure 1. The “File” menu at the top of the UI allows loading and saving of various kinds of files, while the panels below that allow the user to select which

overlays are shown, add and edit holes and slits, select a grism, or adjust the default parameters for the grism.

An object (slit or hole) can be added by clicking precisely within the image, outlining a box around an object to find the center of it, or typing directly in the position in either pixels or celestial coordinates.

6. Advantages of the MOIRCS Mask Builder Plugin

The new plugin offers a number of desirable features over the older IDL program:

- Ginga is open-source and licensed under the generous and unrestrictive BSD license that the rest of the Python ecosystem is based on, so there are no restrictions on where it can be installed and used.
- The instability of the old program is eliminated; Ginga is an extremely stable program that has been in continuous nightly use at the observatory since 2012.
- The plugin allows performant zooming and panning, and allows us to provide a position angle feature so the user can type in a position angle to see the spectral dispersion with various rotations vs. the focal plane.
- The plugin correctly handles overplotting on different pixel scales, so the user should no longer have to provide an image pixel-scaled to an exact value (although we still need to verify that the mask cutting file is generated correctly with alternate scales).

7. Getting and Installing the MOIRCS Mask Builder Plugin

To use the MOIRCS Mask Builder Plugin you need to first install Ginga and then one additional package. We recommend using a virtual python environment such as *virtualenv*, *conda* or similar to install and run this plugin. An example installation is shown below assuming a conda-type system:

```
$ conda env create -n moircs_mdp python=3.13
$ conda activate moircs_mdp
$ pip install ginga[qt6]
$ pip install git+https://github.com/naojsoft/naojutils
```

After installing this way you can run the MOIRCS MBP by simply typing "ginga" on the command line inside your conda environment, and then opening the plugin from the "Plugins => Subaru => Planning => MOIRCS Mask Builder".

References

Ichikawa, T., Suzuki, R., Tokoku, C., Uchimoto, Y., Konishi, M., Yoshikawa, T., Yamada, T., Tanaka, I., Omata, K., & Nishimura, T. 2006, Proceedings of SPIE - The International Society for Optical Engineering, 6269

Jeschke, E., Inagaki, T., & Kackley, R. 2012, in Software and Cyberinfrastructure for Astronomy II, edited by N. Radziwill, & G. Chiozzi (SPIE). Vol. 8451

SOFTWARE, N. G. 2025, The IDL Virtual Machine, `https://www.nv5geospatialsoftware.com/Support/Maintenance-Detail/the-idl-virtual-machine`. Accessed on October 22, 2025

Astronomical Data Analysis Software and Systems XXXV
*ASP Conference Series, Vol. **VOL** Compiled: Tuesday 30th June, 2026@ 01:30*
K. Polsterer, D. Bohlender, S. Gwyn, A. Hincks and P. Teuben, eds.
*© **YEAR** Astronomical Society of the Pacific*

ZZZDEBUG!186.tex
Molinaro, M.|textbf
Allen, M.
Wambsganns, J.
Solano, E.
Cecconi, B.
Demleitner, M.
Schaaff, A.
Heinl, H.
Bertocco, S.
projects!ESCAPE
projects!Astro-CC
projects!EOSC
Virtual Observatory (VO)
International Virtual Observatory Alliance (IVOA)
FAIR

Astronomy Open Science Competence Centre in Europe

Marco Molinaro,[1] Mark Allen,[2] Joachim Wambsganns,[3] Enrique Solano,[4] Baptiste Cecconi,[5] Markus Demleitner,[3] André Schaaff,[2] Hendrik Heinl,[1] and Sara Bertocco[1]

[1]*INAF - Astronomical Observatory of Trieste, Trieste, Italy;* marco.molinaro@inaf.it

[2]*CNRS / CDS, Strasbourg, France*

[3]*University of Heidelberg, Heidelberg, Germany*

[4]*INTA CSIC, Madrid, Spain*

[5]*Paris Observatory, Paris, France*

Abstract. The Astronomy Open Science Competence Centre Pilot (Astro-CC) is an ESCAPE-cluster related project meant to enable the astronomy research communities to accelerate their use of Open Science by supporting the implementation of FAIR principles.

The Astro-CC project aims at expanding the use of Virtual Observatory standards by astronomy-focused ESFRIs, RIs, and data-producing projects of all scales, enabling the astronomy research communities to accelerate their use of Open Science by supporting the implementation of FAIR principles. It will run community events engaging experts in astronomical data & service interoperability to prepare and define the scope of a Community Competence Center.

The project will support the community development of the Virtual Observatory interoperability framework and its integration into EOSC, building on the progress made in the ESCAPE Science Cluster project. It aims at contributing to the vision of EOSC as a federation, providing feedback on the practical implementation of Open Science to the EOSC governance.

1. The Project

The Horizon Europe Open Science Clusters' Action for Reasearch & Society (OSCARS[1]) has a goal to foster the uptake of Open Science in Europe, consolidating past achievements of the thematic Science Clusters into FAIR data service and working practices.

To support the research communities to take up Open Science and foster the involvement of scientists in EOSC, OSCARS provided Open Calls for cascading grants to support researchers in uptaking FAIR principles.

[1]https://oscars-project.eu/

The Astronomy Open Science Competence Centre Pilot project (Astro-CC[2]) is one of the projects funded by the first OSCARS' Open Call, and it is related to the ESCAPE Science Cluster (including astronomy, nuclear and particle physics), continuation of the Horizon 2020 ESCAPE project.

Astro-CC will help the astronomy research communities in the uptake of Open Science and supporting the implementation of FAIR principles. The project will support the community development of the Virtual Observatory interoperability framework and its integration into EOSC. The Virtual Observatory open standards are the basis for making astronomy data FAIR.

The complexity of managing astronomical data across various platforms and ensuring that it adheres to the FAIR principles, requires a community approach. The community also needs training and improved access to tools that support data interoperability and sharing across multiple RIs.

Astro-CC addresses this challenge by running Community Competence Centre events, providing training on the implementation of interoperable services, the development of the underlying Virtual Observatory data sharing framework and its integration in EOSC, and exploitation of the system for scientific research. These events (better described in Sec. 2) will specifically target different facets of the astronomy communities:

- Data Provider Forum event;
- Technology Forum events;
- Scientific Training events.

The project will engage national Research Infrastructures (RIs) and expand into interdisciplinary fields, such as planetary science and heliophysics. These efforts aim to promote interoperability across diverse astronomical domains. Astro-CC activities are aimed at extending the participation beyond the ESCAPE Science Cluster astronomy/astro-particle fields to include the wider astronomy communities of planetary science, heliophysics and space weather.

The community events of this project will engage the European experts in astronomical data/ service interoperability to prepare and define the scope of a Competence Centre that will serve the extended range of astronomical communities.

Astro-CC leverages the opportunities of the OSCARS program to advance the connection of disciplinary interoperability frameworks to EOSC, contributing to the vision of EOSC as a federation, and providing feedback on the practical implementation of Open Science to the EOSC governance.

2. The Events

As mentioned, the Astro-CC events will be specifically targeted to engage different facets of the astronomy communities.

A Data Provider Forum event: for Research Infrastructures and data-intensive projects to share best practices for the implementation of standards to provide services for FAIR data.

[2] https://www.oscars-project.eu/projects/astro-cc-astronomy-open-science-competence-centre-pilot

Technology Forum events: for the developers of astronomical software/services; to discuss community open standards and approaches to FAIR in the astronomy (in its largest definition) research domain.

Scientific Training events: for astrophysics Ph.D students and early career researchers; to learn using interoperable tools and services as well as to gain skills for Open Science publication and research in astronomy. These training events leverage operational services, software and training tutorials that have been on-boarded in EOSC.

At the time of writing a couple of events took already place:

- the 1st Technology Forum[3], held in Trieste in the second week of October 2025;
- the 1st Scientific Training[4], held in Madrid in the first week of December 2025;

while the Data Providers Forum[5] is under organisation and will happen in Heidelberg at the end of March 2026.

A small report of the 1st Tech Forum is available in this proceeding (see Sec. 3). For the training and the providers' forum it's possible to find further information on the events' websites (linked here above from the footnotes).

3. The 1st Astro-CC Tech Forum

The 1st Technology Forum event of Astro-CC took place at the Basovizza Observing Station of the INAF-OATs (Trieste, Italy) with remote participation available for talks and (best effort) hands-on "hack-a-thons". It was attended by 30 participants reporting in talks and/or discussing or doing hands-on work on:

- VO standards' updates;
- Planetary data publishing;
- EPRV spectra;
- FAIRness, semantics and metrics;
- AI relationship to VO standards;
- VO relationship to other platforms;
- Contributing the PyVO effort.

The format balanced 4 plenary sessions for talks with 10 hack-a-thon slots; all the slots where 90' long. An intro session and a wrap up enclosed the nearly 3 days of continuous work.

The forum provided a valuable venue for the community to connect and discuss the topics listed above, clarifyng open points about existing standards, exemplifying

[3] 1st Tech Forum website: `https://indico.in2p3.fr/event/36318/`
[4] 1st Training Event website: `https://indico.in2p3.fr/event/36738/`
[5] Data Providers Forum website: `https://indico.in2p3.fr/event/37439/`

existing technologies, discussing community requirements towards new standardisation needs, and allowed for a place to consider what it means a thematic node in EOSC for the community of astrophysics.

4. Next Steps

The Astro-CC project is moving towards the end of its first year of activities. At the end of the first 12 months, 3 out of the proposed 5 events will be closed and a first report could be done. The remaining part of the project will see the 2nd Technology Forum and the 2nd Scientific Training take place, so far under definition in terms of location and timeframe. One goal of the project itself will be clarifying what would be the goal, benefits, and efforts required in implementing and maintaining a thematic node for astrophysics (at large) within the Federation of the EOSC Nodes.

Acknowledgments. The OSCARS project has received funding from the European Commission's Horizon Europe Research and Innovation programme under grant agreement No.101129751. The Astro-CC project has been funded through the first Cascading Grants call of the OSCARS project.

Astronomical Data Analysis Software and Systems XXXV
*ASP Conference Series, Vol. **VOL** Compiled: Tuesday 30th June, 2026@ 01:30*
K. Polsterer, D. Bohlender, S. Gwyn, A. Hincks and P. Teuben, eds.
*© **YEAR** Astronomical Society of the Pacific*

ZZZDEBUG!187.tex
FernÃ¡ndez, V.|textbf
Espinosa, J.
ArÃ©valo, M.
LÃ³pez-Caniego, M.
Baines, D.
Labiano, A.
FernÃ¡ndez, M.
MerÃn, B.
Sonnentrucker, P.
Evans, C.

Enhancing the ESA JWST Science Archive Ingestion Pipeline

V. Fernández,[1] J. Espinosa,[1] M. Arévalo,[1], M. López-Caniego,[2], D. Baines,[4] A. Labiano[3], M. Fernández,[1] B. Merín,[4] P. Sonnentrucker,[5] and C. Evans,[5]

[1]*Starion Spain for ESA, Madrid, Spain;* `Victor.FernandezRubio@ext.esa.int`

[2]*Aurora B.V. for ESA, Madrid, Spain*

[3]*Telespazio for ESA, Madrid, Spain*

[4]*ESA, ESAC, Madrid, Spain*

[5]*ESA, ESA Office, Space Telescope Science Institute, Baltimore, USA*

Abstract.

The ESAC Science Data Centre (ESDC) provides the astronomical community with open access to data from a wide range of space missions, including the James Webb Space Telescope (JWST). To improve performance, scalability, and interoperability, ESDC has implemented a new ingestion workflow for JWST products based on icewind, a high-performance data transfer and ingestion application developed by the Canadian Astronomy Data Centre (CADC). This transition to icewind dockerized deployment, together with additional features integrated by ESDC, has significantly increased ingestion speed and operational efficiency, while reducing maintenance overhead.

In addition to performance gains and thanks to small contributions from ESDC to this software, icewind now enables the retrieval and integration of High-Level Science Products (HLSPs) from the Mikulski Archive for Space Telescopes (MAST) into the ESDC Science Archives. A first subset of these products are now accessible through the ESA JWST Science Archive, expanding scientific reach and enhancing data discoverability. This work demonstrates the benefits of adopting containerized solutions for large-scale astronomical data systems as well as the tangible outcomes of close international collaboration..

1. Introduction

The James Webb Space Telescope (JWST) is delivering a vast volume of high-quality astronomical data, covering a wide range of instruments, observing modes, and scientific use cases. Ensuring timely and reliable access to these data products through science archives requires ingestion pipelines that are not only performant, but also scalable, maintainable, and resilient to operational failures.

The ESA JWST Science Archive [1] [Arévalo et al. (2025), Labiano et al. (2023)], operated at the ESAC Science Data Centre (ESDC), is responsible for the continuous ingestion, validation, and serving of JWST products. This task has become increasingly

[1]Latest version of eJWST interface: https://jwst.esac.esa.int/ejwst

challenging as data volumes grow and as new types of externally generated products are incorporated. In particular, the integration of High-Level Science Products (HLSP), provided to MAST by external scientists, introduces additional heterogeneity in data sources, metadata structures, and update cycles. Scalability, reliability, and long-term maintainability become new challenges, making existing approaches insufficient for the demands of modern astronomical archives.

To address these issues, the CADC icewind framework [Redman & Dowler (2013). Redman & Dowler (2013)] was adopted and extended within the ESA JWST Science Archive, with several operational adaptations and added support for HLSP ingestion. This work resulted in improved ingestion performance and robustness, and enabled the successful integration of JWST HLSPs from MAST [Fleming et al. (2015)] into the ESA Science Archives.

2. Background and Motivation

Modern astronomical archives increasingly rely on distributed data sources, automated pipelines, and close collaboration between institutions. As flagship missions enter full operational phases, such as JWST in 2022, archives must handle sustained high data rates while ensuring metadata integrity and long-term preservation.

Icewind is a high-performance ingestion and synchronization framework developed by the CADC to address these requirements. It combines parallel data transfer, configurable execution modes, and container-based deployment to efficiently synchronize metadata and data products between remote services and local archives. Its modular design allows the same framework to be reused across different missions and archives, reducing development and maintenance effort.

In parallel, the scientific exploitation of JWST data has expanded through the production of HLSPs. These science-ready datasets, generated by expert teams and distributed via the MAST Science Archive [Shaw et al. (2025)], include images, spectra, catalogs, and derived products based on reprocessed observations from a variety of missions. Integrating selected HLSP collections into the ESA JWST Science Archive enhances scientific reach and provides users with a more complete and interoperable data discovery experience.

However, ingestion of HLSP poses additional challenges compared to mission pipeline products, including external update cycles, heterogeneous metadata, and the need for robust validation mechanisms. The collaboration between ESDC, MAST and CADC provided an opportunity to extend the icewind framework beyond its original scope, enabling HLSP mirroring while preserving operational stability and compliance with ESDC standards.

3. System Overview

The adopted solution is based on a fully containerized deployment of the **icewind** framework, extended with ESDC-specific adaptations. The overall approach aims to encapsulate the complete ingestion logic, runtime dependencies, and execution workflow into a set of reproducible container images, minimizing host-level dependencies and simplifying operational deployment.

At a high level, the system integrates the Java-based ingestion modules provided by CADC with additional customization layers developed at ESDC. These layers adapt configuration management, execution control, and security aspects to comply with internal operational standards. This application is built and deployed in an image. The resulting container where this image is executed can be instantiated multiple times with different configurations at runtime, enabling distinct operational behaviors without modifying the underlying software artifact.

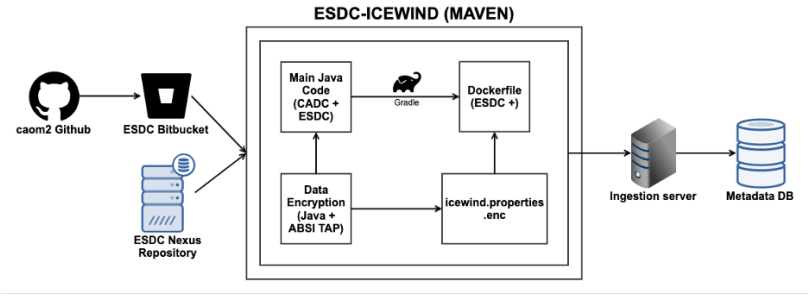


Figure 1. High-level overview of the icewind-based ingestion architecture at ESDC.

The ingestion workflow is divided into two main phases:

- **Metadata synchronization:** the application incrementally queries the MAST service to identify new or updated observations. This phase supports multiple execution modes and different collections to address routine operations, reprocessing scenarios, and consistency checks. There is a set of execution modes defined:
  - **Standard mode:** Performs incremental harvesting and ingestion of new or updated products.
  - **Skip mode:** Allows operators to reprocess previously failed observations without restarting full ingestion cycles.
  - **Validate mode:** Compares local and remote metadata, supporting consistency checks and quality control activities.
- **Data download:** The framework retrieves the files associated to the newly synchronized observations and organizes them within the local archive structure.
  - **Discover:** Analyses the latest metadata and adds the available files in a queue list.
  - **Download:** Downloads the files discovered in the previous step.

This architecture enables parallel execution, robust error handling, and repeatable operations, which are essential for managing the growing volume and diversity of JWST data products. On the other hand, these execution modes, combined with containerized deployment, significantly reduce manual intervention, improve fault tolerance, and simplify long-term maintenance of the ingestion pipeline.

4. Conclusions and Future Work

The adoption of the icewind framework has modernized the JWST ingestion pipeline at ESDC, improving scalability, reliability, and operational efficiency. The successful integration of HLSPs from MAST expands the scientific scope of the ESA JWST Science Archive and strengthens international collaboration. The icewind-based ingestion workflow is currently used in operations to synchronize JWST metadata and data products, including selected HLSP collections retrieved from MAST. The containerized approach has simplified deployment and maintenance while enabling parallel execution and improved fault tolerance.

Operational experience shows a significant reduction in ingestion turnaround time and manual intervention, together with increased robustness when handling partial failures or external service interruptions. Future work includes extending this approach to additional missions and Science Archives, such as the ESA Hubble Science Archive (eHST[2]), and product types, as well as further automation of validation and monitoring capabilities within the ingestion workflow.

References

Arévalo, M., Espinosa, J., & López-Caniego, M. 2025, in Astronomical Society of the Pacific Conference Series, edited by A. Jacques, R. Seaman, N. Gandilo, & T. Linder, vol. 541 of Astronomical Society of the Pacific Conference Series, 52

Fleming, S. W., Abney, F., Donaldson, T., Dower, T., Fraquelli, D. A., Koekemoer, A. M., Levay, K., Matuskey, J., McLean, B., Quick, L., Rogers, A., Shiao, B., Thompson, R., Tseng, S.-A., Wallace, G., & White, R. L. 2015, in American Astronomical Society Meeting Abstracts #225, vol. 225 of American Astronomical Society Meeting Abstracts, 336.59

Labiano, A., Marston, A., López-Caniego, M., Arévalo, M., Espinosa, J., & Ballesteros, J. 2023, in Highlights on Spanish Astrophysics XI, edited by M. Manteiga, L. Bellot, P. Benavidez, A. de Lorenzo-Cáceres, M. A. Fuente, M. J. Martínez, M. Vázquez Acosta, & C. Dafonte, 401

Redman, R. O., & Dowler, P. 2013, in Astronomical Data Analysis Software and Systems XXII, edited by D. N. Friedel, vol. 475 of Astronomical Society of the Pacific Conference Series, 159

Shaw, R. A., Novacescu, J. L., Weissman, S., Berger, T. A., Brasseur, C. E., Chamblee, J., Cherinka, B., Claytor, Z. R., Dower, T., Edeani, C., Fleming, S. W., Hargis, J. R., Imig, J., Januario, T., Levay, K., Kimball, T., Kotler, J., Lewis, H. M., Lubow, S., Lucy, A., McLean, B., Malla, S. G., Matuskey, J., Miller, S. J., Mullally, S. E., Murray, C. E., Peek, J. E. G., Phillip, C., Rafelski, M., Rodriguez, D. R., Snyder, G. F., Usha, A. J., White, R. L., & Yoon, J. 2025, arXiv e-prints, arXiv:2512.18101. eprint: 2512.18101

[2]https://hst.esac.esa.int/ehst

Astronomical Data Analysis Software and Systems XXXV
*ASP Conference Series, Vol. **VOL** Compiled: Tuesday 30th June, 2026@ 01:30*
K. Polsterer, D. Bohlender, S. Gwyn, A. Hincks and P. Teuben, eds.
*© **YEAR** Astronomical Society of the Pacific*

ZZZDEBUG!190.tex
Savchenko, V.|textbf

Assembly, Integration, and Verification of the Cherenkov Telescope Array Observatory Data Processing and Preservation System

Volodymyr Savchenko, on behalf of the CTAO DPPS team and SUSS AIV team

[1]*EPFL, Lausanne, Switzerland;* Volodymyr.Savchenko@epfl.ch

Abstract.
The Cherenkov Telescope Array Observatory (CTAO) is the next-generation very-high energy gamma-ray observatory currently under construction. The bulk of the CTAO data will be processed by the Data Processing and Preservation System (DPPS), yielding high-level data ready for elaboration by astronomers. Assembly, Integration, and Verification (AIV) is a key element of the DPPS development lifecycle, integrating its subsystems into a cohesive system, and verifying that it meets its requirements and quality standards during development and deployment, as well as creating the means to ensure its long-term operability and maintainability.

DPPS AIV embraces the synergy of dynamic Continuous Delivery and rigorous Requirements Verification workflows, built upon DevOps principles enabling reproducible deployments closely aligned between development, testing, pre-production, and production environments as well as reliable fail-over between different sites of the distributed production infrastructure.

1. CTAO and DPPS

The Cherenkov Telescope Array Observatory (CTAO) represents the next generation of ground-based gamma-ray astronomy. Currently under construction at two sites – La Palma, Spain (North) and Paranal, Chile (South), the observatory aims to detect high-energy photons in the energy range from around 20 GeV to 300 TeV with unprecedented sensitivity, angular resolution, and energy resolution.

The bulk of the CTAO data will be processed by the Data Processing and Preservation System (DPPS) [1], yielding high-level data ready for elaboration by astronomers. DPPS enables **Data Preservation** for decades of the observatory lifetime with multi-site redundancy, **High-Volume Data-Driven Processing** by executing complex data-driven workflows converting raw telescope data into science-ready data products.

The DPPS the following management systems: **WMS** – Workload Management System, based on DIRAC and CVMFS (**??**) for workload orchestration and distributed computing ; **BDMS** – Bulk Data Management System, based on Rucio (**?**), for data lifecycle management across multiple geographically distributed data centers; and **Ops** – for deployment, and observability of DPPS services, based on Kubernetes, Helm, and Grafana observability stack.

[1] https://gitlab.cta-observatory.org/cta-computing/dpps/dpps/

DPPS orchestrates data processing pipelines, grouped into four subsystems: **CalibPipe** handles telescope and ancillary instrument calibration, **DataPipe** performs event reconstruction, **SimPipe** produces Monte Carlo simulations, and **QualPipe** ensures data quality.

2. DPPS deployment strategy

The deployment strategy lays the foundation of DPPS verification and release lifecycle by defining the way DPPS is brought to life in various environments. It is built on the idea of **reproducible deployment**, allowing to easily create full copies of the DPPS system in any standard Kubernetes cluster.

DPPS Helm charts enable reproducible deployments in any standard Kubernetes cluster. The DPPS system includes its components as Helm chart dependencies. All DPPS Helm charts include test versions of all dependent services.The charts also take care of bootstrapping, upgrades (including migrations), backups and restore.

DPPS CWL pipelines (**?**) are used to define containerized data processing pipelines executed by DPPS WMS since Kubernetes native Job management is not currently suitable for defining complex data processing workflows. DPPS pipelines can be executed in any CWL-compliant execution environment, local or distributed.

All DPPS release artifacts are versioned and stored in an OCI-compliant Harbor (`https://goharbor.io/`) container registry. All DPPS components can be deployed in multiple **deployment environments**. **Local development** and **CI/CD ephemeral environments** are ephemeral environments created on-demand in developer workstations. **Staging**, **Pre-production**: and **Production** are persistent deployment environments managed as GitOps repositories.

Observability is a key aspect of the DPPS deployment strategy, allowing to design, monitor, and verify DPPS deployments at every stage of the deployment lifecycle, producing standardized metrics, logs, and alerts, which can be interrogated and analyzed in the same way as full-scale pre-production and production deployments.

3. Release Lifecycle and Verification Strategy

DPPS embraces a rapid, trunk-based development cycle, with relatively long sprints (about eight weeks) each concluding with a DPPS system release. AIV is central in this process by providing the means to verify DPPS components and system as a whole at every stage of development and deployment.

CTAO, the stakeholder of DPPS, is defining high-level requirements and use cases in the Jama (`https://jama.cta-observatory.org/`) requirements management system. DPPS AIV ensures that every DPPS release is compliant with its requirements by executing automated tests in the CI/CD pipelines upon every code change.

Given that containerized deployment can be easily ported to any standard environment, the main deployment challenge is to ensure that all **system interfaces** are correctly implemented and configured. In addition to scenario-driven integration tests, we test configurability by introducing randomized configuration of interfaces.

Large-scale performance tests are only performed in **pre-production or production** infrastructure, first developed just like other automated tests, and later configured

for pre-production environment. These will continue to be used to verify ongoing compliance of live DPPS deployment with its requirements.

All DPPS code repositories undergo static code analysis using SonarQube (`https://sonar-ctao.zeuthen.desy.de/`) to ensure code quality, maintainability, and security. AIV also collects information about third-party dependencies used by DPPS components, verifying their licenses and known vulnerabilities.

4. CTAO AIV toolkit

In order to efficiently manage the complexity of DPPS AIV processes and tools, we developed a custom toolkit[2] to automate and streamline various AIV tasks. Originally developed to support DPPS, it is now also adopted by another CTAO software project, SUSS. The toolkit includes a **Python Package**, **CI/CD Pipeline Templates** for consistent testing and deployment across projects, a **pytest plugin**: for annotating tests with requirement traceability information, an **AIV Toolkit Helm Chart** for deploying services facilitating testing, and a **pre-commit Hook**.

Before developing our own toolkit, we evaluated existing solutions for setting up a Helm-based development environment, such as skaffold (`https://skaffold.dev/`), and chartpress (`https://chartpress.readthedocs.io/`), but found that they did not fully meet our specific needs, and are difficult to extend to support requirement traceability.

To automate AIV tasks not triggered by code changes, we developed an **AIV Bot**[3], taking care of **release issue management**, **dependency updates**, and integration between Gitlab issue code with **Jama requirements** management system. The AIV bot also contains experimental functionality for generative LLMs to assist with some of its tasks, such as drafting issue descriptions or summarizing changes and comparing them to requirements.

5. LLMs in DPPS AIV

The advent of LLMs has introduced both opportunities and risks to the CTAO software stack. While freeing from certain tasks, it also introduces suboptimal or incorrect code that may pass superficial reviews. Since DPPS AIV is tasked with verifying software quality, it plays a crucial role in mitigating AI-related risks. For example, AI can assist in generating test cases based on requirements, or in analyzing test results to identify patterns or anomalies that may indicate deeper issues.

6. Challenges, Conclusions, and Outlook

Since the first DPPS release in February 2025, DPPS has undergone six minor releases, each verified against a progressively larger set of requirements. Only six months after the first release, the DPPS successfully made first pre-production deployment, and underwent several tests on real data acquired from on-site telescopes, demonstrating

[2] `https://gitlab.cta-observatory.org/cta-computing/common/aiv-toolkit`
[3] `https://gitlab.cta-observatory.org/cta-computing/dpps/aiv/gitlab-operator/`

that DPPS AIV strategy is effective in ensuring that DPPS meets its requirements and quality standards before deployment.

As DPPS development progresses the entire system becomes increasingly complex, with more components, interfaces, and requirements to manage. The current DPPS version comprises almost a hundred containers, dozens of Helm charts, and hundreds of automated tests. The AIV team is continuously adapting to this growing complexity, helping to ensure that the developers can focus on building new features while maintaining confidence in DPPS releases being production-ready.

Acknowledgments. This work was conducted in the context of the CTAO DPPS and SUSS AIV teams. The author thanks the DPPS and SUSS team members for fruitful discussions and feedback. The content of this paper was produced with limited support of several LLMs for drafting and editing assistance.

Astronomical Data Analysis Software and Systems XXXV
*ASP Conference Series, Vol. **VOL** Compiled: Tuesday 30th June, 2026@ 01:30*
K. Polsterer, D. Bohlender, S. Gwyn, A. Hincks and P. Teuben, eds.
*© **YEAR** Astronomical Society of the Pacific*

ZZZDEBUG!191.tex
Li, X.|textbf
Stolyarov, V.
FOOBAR!conference!ADASS 2019
FOOBAR!organisations!ASP
FOOBAR! ascl:1101.010

PulSKASim: A Pulsar Simulator for SKA-Scale Interferometric Observations

Xiaotong Li[1] and Vladislav Stolyarov[2]

[1]*University of Oxford, Oxford, United Kingdom;* egbdfmusic1@gmail.com

[2]*University of Cambridge, Cambridge, United Kingdom;* vs237@cam.ac.uk

Abstract. Accurate simulation of pulsar flux variability is critical for testing Square Kilometre Array (SKA) interferometric pipelines. However, most existing simulators neglect the effects of integration time and related observational parameters, limiting their realism and utility for interferometric end-to-end testing. To address these shortcomings, we develop a Pulsar Simulator for SKA-scale interferometric observations (PulSKASim), which models pulsar flux evolution across pulsar period, maximum flux, duty cycle, and noise, accounting for integration time, sampling, and observation duration, and naturally models flux smoothing that arises from finite integration within each dump time. PulSKASim generates synthetic measurement sets using functions in simulators for radio interferometers, such as OSKAR and Pyuvsim, where each snapshot contains pulsars with controlled flux levels, enabling realistic per-time-slot experiments. This simulator allows for detailed assessment of calibration, imaging, and detection pipelines under realistic SKA-like conditions, bridging pulsar variability modelling with interferometric simulation in a way not achievable by existing tools.

1. Introduction

Pulsars are rotating, highly magnetised neutron stars whose beams produce pulsing signals to observers on Earth. They are valuable probes of strong-field gravity, ultra-dense matter, and the interstellar medium, and form a core component of the Square Kilometre Array (SKA) science programme[1]. Accurate pulsar simulations are essential for designing observing strategies, validating signal-processing pipelines, and preparing for next-generation radio telescopes, such as the SKA. Existing pulsar simulators, such as PsrSimSig[2] and Radio Pulsar Signal Generator (Santos et al. 2018), are not designed for producing interferometric Measurement Sets (MSs) with pulsar variability.

To fill this gap, we develop a simulator, called PulSKASim, which supports SKA-scale configurations and enables testing of interferometric pulsar search, imaging, and timing pipelines under realistic conditions. PulSKASim is open source and is available on GitHub [3].

[1]https://www.skao.int/en/explore/science-goals
[2]https://psrsigsim.readthedocs.io/en/latest/
[3]https://github.com/egbdfX/PulSKASim/

2. Methodology

The structure of PulSKASim is shown in Fig. 1. It consists of two major components, a flux generator and a radio interferometric (RI) simulator. The flux generator produces a flux sequence with one value per snapshot, given the pulsar period T, maximum pulsar amplitude A, sampling period (telescope dump time) T_s, pulsar duty cycle D, total simulated duration $T_{\rm sim}$, and noise standard deviation σ_n. This sequence is inputted into an RI simulator, such as OSKAR[4] (Mort et al. 2010) or Pyuvsim[5] (Lanman et al. 2019). Simulated MSs are generated by specifying the observing frequency, phase centre, pulsar position, observation time, dump time, and telescope layout to the RI simulator.

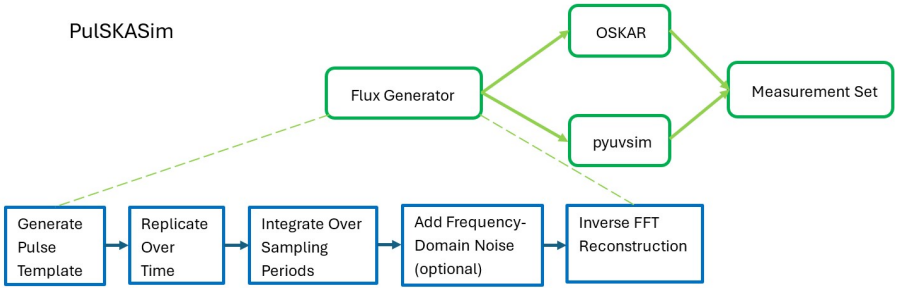


Figure 1. Structure of PulSKASim.

Specifically, the flux generator first creates a pulsar template, modelling the pulsar profile as a Gaussian (Lorimer & Kramer 2004) with width determined by D. The pulsar template is offset-corrected so that it smoothly connects to zero outside the duty cycle and is then replicated over the total simulated duration to produce a continuous periodic signal. To mimic interferometer integration, instantaneous brightness is integrated over each sampling period to produce time-averaged flux values. Optionally, to simulate realistic noise, random noise is introduced into the Fourier domain, controlled by σ_n. The flux is then reconstructed in the time domain via inverse Fourier transform, and the resulting sequential flux values are stored in the output catalogue.

The flux generator subsequently triggers the RI simulator. In this work, we consider OSKAR, a GPU-accelerated RI simulator optimised for large arrays such as the SKA. For time-variable flux realisations, each snapshot is simulated as an individual MS. Each simulation can be run in parallel across multiple allocated nodes. The resulting snapshot-level datasets can then be merged into a single MS using *casacore* functions. The merged MS is the output of PulSKASim. Alternatively, we employ Pyuvsim, a lightweight, Python-based simulator designed for accurate visibility calculations and algorithm testing in radio astronomy. Similar to OSKAR, to simulate time-varying sources by Pyuvsim, users must run separate simulations for each time step with updated sky models; these runs can be parallelised with MPI, and the resulting datasets (usually in *uvh5* format) should be merged afterwards using *pyuvdata* or *casacore* and converted to the MS.

[4] https://ska-telescope.gitlab.io/sim/oskar/
[5] https://pyuvsim.readthedocs.io/en/latest/index.html

3. Performance

3.1. Fidelity

The fidelity of PulSKASim depends largely on the performance of the flux generator. To assess it, we compare the simulated pulsar flux with the observation of PSR J0901-4046 (Caleb et al. 2022) as an example. A total of 1500 snapshots were generated in this experiment. Figure 2 (Left) shows the frequency spectrum of the real pulsar alongside the noise-free simulated spectrum. In it, the real signal exhibits a prominent low-frequency component, which corresponds to low-frequency noise. This is evident in Fig. 2 (Middle), where removing the low-frequency component reveals the high-pass filtered signal. Furthermore, as shown in Fig. 2 (Right), by adding the same noise pattern as the real signal, the frequency spectrum of the simulated signal closely matches that of the real signal, demonstrating the high fidelity of the simulation.

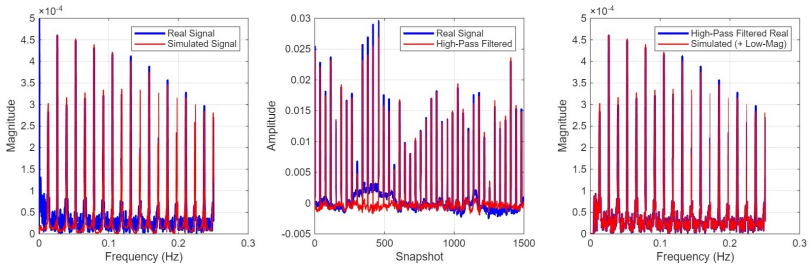


Figure 2. Fidelity assessment. *Left:* Frequency spectra of PSR J0901-4046 and the noise-free PulSKASim simulation (T = 75.886s, T_s = 1.999s, D = 1.4/360, $T_{\rm sim}$ = 3000s, σ_n = 0.0). *Middle:* Comparison of the original and high-pass filtered signals, showing the effect of low-frequency removal. *Right:* Frequency spectrum of the simulated signal with real-noise pattern compared to that of the real signal.

3.2. Robustness

The robustness of PulSKASim is evaluated under different Nyquist sampling conditions by varying the ratio between T_s and T. The results indicate that the simulator behaves reliably across oversampling, critical sampling, and undersampling cases. PulSKASim is a radio interferometric simulator designed for pulsar observations. It can be used to generate synthetic data for testing Fast Imaging Pipelines (FIPs) (Stolyarov et al. 2025; Li et al. 2025). Such image-domain detection pipelines are particularly effective for identifying long-period transients, which are challenging to detect using traditional time-domain approaches. In this case, the transients are typically oversampled. Furthermore, these FIPs can also be applied to short-period transients: although undersampling introduces frequency aliasing, the transients can still be successfully localised in the image domain. Consequently, the robustness of PulSKASim ensures that it is applicable to testing both oversampled and undersampled science cases.

3.3. Computational Efficiency

To evaluate the computational efficiency of PulSKASim, we perform simulations using the parameters of PSR J2251-3711 (Morello et al. 2020) and the MeerKAT telescope model as an example. The median time per snapshot for simulations with different number of frequency channels is shown in Fig. 3. Since Pyuvsim is designed as reference

software that prioritises high accuracy, it is not expected to deliver high computational performance in simulations. It relies on MPI for parallelisation, and its execution time depends significantly on the number of CPU cores used. We tested both simulators on a consumer desktop equipped with a 6-core AMD Ryzen 5600X CPU and an NVIDIA RTX 3060 GPU. Our tests show that, to match the performance achieved by OSKAR using a GPU with approximately 3000 CUDA cores, Pyuvsim must be run on an HPC compute node with 100-200 CPU cores.

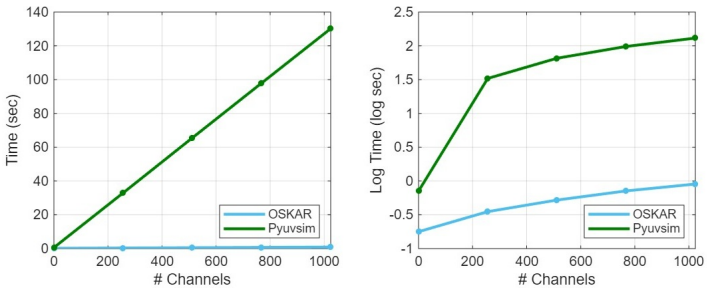


Figure 3. Computation times in (*Left*) linear and (*Right*) logarithmic representations.

4. Conclusions

We propose PulSKASim, an open-source simulator capable of producing realistic, time-variable pulsar signals in interferometric measurements. Our results show high fidelity to real pulsar observations, robustness across a range of sampling conditions, and scalable computational performance. PulSKASim fills a critical gap in pulsar research, providing a tool for designing observing strategies, validating pipelines, and preparing for next-generation radio telescopes.

Acknowledgments. The authors would like to acknowledge the use of the ATNF Pulsar Catalogue[6] (Manchester et al. 2005).

References

Caleb, M., Heywood, I., Rajwade, K., et al. 2022, Nat. Astron., 6, 828
Lanman, A. E., Hazelton, B. J., Jacobs, D. C., Kolopanis, M. J., Pober, J. C., Aguirre, J. E., & Thyagarajan, N. 2019, J. Open Source Softw., 4, 1234
Li, X., Adamek, K., Giles, M., & Armour, W. 2025, FIP-TOI: Fast imaging pipeline for pulsar localisation with a transient-oriented radio astronomical imager. eprint: 2512.06254
Lorimer, D., & Kramer, M. 2004, Handbook of Pulsar Astronomy, Cambridge Observing Handbooks for Research Astronomers (Cambridge University Press)
Manchester, R. N., Hobbs, G. B., Teoh, A., & Hobbs, M. 2005, AJ, 129, 1993
Morello, V., Keane, E. F., Enoto, T., et al. 2020, MNRAS, 493, 1165
Mort, B. J., Dulwich, F., Salvini, S., Adami, K. Z., & Jones, M. E. 2010, in IEEE PAST, 690
Santos, J., Brito, D., Tavares, G., & Fernandes, J. 2018, in 2018 IEEE ISCAS, 1
Stolyarov, V., Li, X., Heywood, I., & Adamek, K. 2025, in PoS(HEASA2025), vol. 514, 009

[6] https://www.atnf.csiro.au/research/pulsar/psrcat

Astronomical Data Analysis Software and Systems XXXV
*ASP Conference Series, Vol. **VOL** Compiled: Tuesday 30th June, 2026@ 01:30*
K. Polsterer, D. Bohlender, S. Gwyn, A. Hincks and P. Teuben, eds.
*© **YEAR** Astronomical Society of the Pacific*

ZZZDEBUG!196.tex
Taylor, M. B.|textbf
data formats
data!tabular

Combining Data and Metadata: Hybrid Tabular Data Formats

Mark Taylor

H. H. Wills Physics Laboratory, Tyndall Avenue, University of Bristol, UK;
m.b.taylor@bristol.ac.uk

Abstract. When working with astronomical data, metadata is also important. A general-purpose file format for transmission, processing and archiving large datasets should facilitate, among other things, both efficient processing of bulk data and encoding of rich semantic metadata. When choosing a format for a particular purpose sometimes no existing format satisfies both these requirements adequately, but combining one data-efficient and one metadata-rich format can be made to do so.

This paper discusses considerations for designing such hybrid data/metadata formats, and reviews some examples such as VOParquet, FITS-plus and ECSV. We focus on tabular data, but some of the considerations may apply to other datatypes such as arrays as well.

1. Introduction

Many considerations go into selecting a suitable file format for a given purpose when working with astronomy data. These include availability of compatible software, ease of use by machines, ease of use by humans, efficiency of processing in one or several modes, storage size, flexibility, support for required data types and ability to represent rich metadata.

Ideally, one existing file format presents itself as satisfying all the required criteria. Where that is not the case however, compromises may be made on one or several requirements, or the decision may be taken to invent a new format or to adapt an existing one.

One approach in this situation is to take a format that provides some of the required functionality and combine it somehow with another format that supplies the missing pieces. There are several examples in current usage where such a "hybrid" format has been defined for use with tabular data in astronomy, in particular where an efficient but metadata-poor format for storage of bulk tabular data (lacking for instance column units and descriptions) is combined with a metadata-rich but less performant format for storage of the associated per-column and per-table semantic metadata, along with some arrangement to associate the data and metadata parts together.

We discuss here the development and use of such hybrid data/metadata formats. Section 2 gives an overview of design considerations, section 3 explains the options for associating the data and metadata parts, and section 4 lists some examples of this approach in practice.

2. Design Considerations

When designing a hybrid data/metadata format, a number of considerations must be taken into account.

Software support is clearly important; it may be possible to work to some extent with the resulting format using standard software intended for one or other of the combined formats, or it may be necessary to write new I/O code from scratch. For writing the hybrid format, and for reading the data and metadata together, at least some modifications to standard software are likely to be required. If the combined format remains comprehensible to off-the-shelf software, for instance if standard tools that understand the data format can be used even though they don't take account of the additional metadata in the hybrid format, utility of the hybrid format is increased.

The data format must be capable of representing the required data, and the metadata format must be capable of representing the required metadata. Additionally a good match, or at least compatibility, between these formats is desirable. For instance if two formats that both annotate datatypes are combined, problems can arise if one of the formats supports some datatypes that the other does not.

It may also be desirable to make the data or metadata readily readable or editable by humans.

3. Data/Metadata Association

The chief technical issue in designing a hybrid format is to define the means by which the metadata and data are joined into a single unit that can be handled by I/O software. This is usually done by adapting one of the formats to "wrap" an instance of the other. This can be done either way round.

In some cases, instances of the hybrid format are instances of the data format, but with some additional metadata (encoded in the metadata format) stored somewhere in the data format that does not disrupt or invalidate it. For example a VOParquet file is a Parquet file with a VOTable header stored in the key-value storage area of its footer.

In other cases, instances of the hybrid format are instances of the metadata-rich format, but with some arrangement to store bulk data in the data-efficient format instead of the usual bulk storage mode. For example a FITS-serialized VOTable is a VOTable file, but the element which would normally encode the bulk data in XML elements or a native VOTable binary stream instead encodes it as a FITS file or binary stream.

If done carefully, this wrapping can be done without disruption to the "wrapper" format, so that such files can be manipulated by standard software unaware of the hybrid nature of such file instances.

This wrapping can be done by embedding or reference. FITS-serialized VOTable for instance allows both options: the VOTable STREAM element that carries the FITS data may either include the FITS binary stream as a base64-encoded child text node, or it may reference an external resource containing the FITS payload. While referencing an external file facilitates manipulation of the the data and metadata separately, it incurs the responsibility to keep the two files or resources together which can be problematic; in most cases the embedding option is more convenient and robust.

4. Examples

We review here some examples of hybrid table data formats in current use in astronomy. Section 4.1 lists some of the data- and metadata-oriented formats in use, and section 4.2 discusses how these have been put together to form working hybrid data formats.

4.1. Example Component Formats

The following are some formats suited to storage of bulk data:

Parquet Apache Parquet is a modern industry-standard format optimised to allow fast and parallel processing of large and very large tables. Many off-the-shelf big data tools can work with it. It has extremely limited standard semantic metadata, but a key-value list allows insertion of custom metadata items.

FITS The Flexible Image Transport System (Pence et al. 2010) is a venerable binary format for storage of astronomy array and tabular data. It is ubiquitous in astronomy, lean, efficient, and easy to implement, but clunky, old-fashioned, not suited to parallel processing, and has restrictive arrangements for metadata.

CSV Comma-Separated Values is a text-based table format. It can be read and written almost anywhere, but is inefficient and has no facilities for metadata storage beyond column name.

The following are capable of storing rich metadata:

VOTable VOTable (Ochsenbein et al. 2025) is an IVOA standard XML-based format for tables in astronomy. It can annotate tables and columns with units, UCDs, coordinate system information, DataLink service descriptors, and other items found to be valuable in the context of the Virtual Observatory.

YAML YAML Ain't Markup Language is a general purpose text-based format capable of storing hierarchical information in a human- and machine-readable form.

4.2. Example Hybrid Formats

The data storage capabilities of FITS have been combined with the metadata encoding capabilities of VOTable in more than one way. FITS and VOTable datatypes are compatible by design since VOTable was intended specifically to represent the same data as FITS binary tables but with enhanced metadata capabilities, as well as lifting some other restrictions. From its inception VOTable has had the option of storing its data in an embedded or referenced FITS stream with description supplied by VOTable metadata. However, this "FITS-serialized VOTable" format is rarely used in practice; if the FITS file is associated by reference the two files are liable to get separated, but the alternative of embedding the FITS stream using base64 in the VOTable XML document makes it difficult to take advantage of the efficiency of the FITS binary format. An alternative known as "FITS-plus" which reverses the direction of wrapping stores a data-less VOTable document encoding metadata only in the primary HDU of a FITS file, so that the FITS file can be manipulated using normal FITS readers, but FITS-plus-aware readers can benefit from the VOTable metadata if it is present. FITS-plus is not currently a public standard, but has long been in use as a private convention by the STIL library (Taylor 2005), and hence the TOPCAT and STILTS applications.

VOTable metadata has also been used in recent years to enhance the Parquet data format, since Parquet is gaining considerable popularity for storage of large astronomy tables but lacks even basic semantic metadata such as column units. Like VOTable/FITS, the VOTable/Parquet hybrid also exists in two forms. The VOTable I/O library in Astropy (Astropy Collaboration et al. 2022) supports a format called `votable.parquet` which references an external Parquet file from a metadata-only VOTable document, in the same way as FITS-serialized VOTable; however this is outside of the VOTable standard so cannot be read by standards-compliant VOTable readers, and it also suffers the inconvenience of splitting the information between two files. A more recent initiative is VOParquet (Taylor et al. 2025) which like FITS-plus stores a metadata-only VOTable in a permitted location in a normal Parquet file (the key-value store in its footer). VOParquet is being adopted by a number of projects and forms part of the evolving HATS framework (Caplar et al. 2025) for efficient storage of all-sky datasets using structured Parquet filesets. The most commonly used Parquet datatypes map straightforwardly on to VOTable equivalents, but Parquet files can exist which cannot be accurately modeled with VOTable metadata, so there are restrictions in the applicability of this pairing.

YAML has also been used to annotate metadata-poor formats. ECSV (Enhanced Character Separated Values) prepends a YAML header to a CSV file to provide additional metadata; the resulting file is neither YAML nor CSV so requires custom software to read, but this can be fairly easily assembled given existing YAML and CSV parsers. MAML is a YAML-based table metadata format that can be stored in the footer of a Parquet file similarly to how VOParquet uses VOTable. Since YAML doesn't describe a table format on its own there is no problem with datatype mismatches, but this means some custom convention has to be established for YAML encoding of the per-table and per-column metadata.

5. Conclusion

The combination of one data-efficient and one metadata-rich format has been made several times in astronomy to define a hybrid storage format for tabular data. While not the most elegant approach, this can provide a convenient and effective way to get the best of both worlds without having to design a new data format from scratch.

References

Astropy Collaboration, et al. 2022, ApJ, 935, 167. eprint: 2206.14220

Caplar, N., DeLucchi, M., & Jurić, M. 2025, Hats: A standard for the hierarchical adaptive tiling scheme in the virtual observatory, IVOA Note 22 August 2025. URL `https://www.ivoa.net/documents/Notes/HATS/`

Ochsenbein, F., et al. 2025, VOTable Format Definition Version 1.5, IVOA Recommendation 16 January 2025

Pence, W. D., Chiappetti, L., Page, C. G., Shaw, R. A., & Stobie, E. 2010, A&A, 524, A42

Taylor, M., Pineau, F.-X., Sipőcz, B., & Dubois-Felsmann, G. 2025, Parquet in the VO, IVOA Note 16 January 2025. URL `https://www.ivoa.net/documents/Notes/VOParquet/`

Taylor, M. B. 2005, in ADASS XIV, edited by P. Shopbell, M. Britton, & R. Ebert, vol. 347 of ASP Conf. Ser., 29

Astronomical Data Analysis Software and Systems XXXV
*ASP Conference Series, Vol. **VOL** Compiled: Tuesday 30th June, 2026@ 01:30*
K. Polsterer, D. Bohlender, S. Gwyn, A. Hincks and P. Teuben, eds.
*© **YEAR** Astronomical Society of the Pacific*

ZZZDEBUG!198.tex
Stoehr, F.|textbf
Farago, A.
Curiban, S.
Manning, A.
Garcia, J.
Hsieh, P.-Y.
Lipnicky, A.
Plunkett, A.
archives!individual!ALMA Science Archive
algorithm!machine learning!deep learning
algorithm!machine learning!unsupervised
algorithm!k-d trees
software!image processing
Big Data
CARTA! ascl:2103.031

The ALMA morphological image simlarity search is now online

Felix Stoehr[1], Andrea Farago[1], Stefan Curiban[1], Alisdair Manning[1], Jorge Garcia[2], Pei-Ying Hsieh[3], Andy Lipnicky[4], and Adele Plunkett[4]

[1]*European Southern Observatory (ESO), Garching, Germany;* `fstoehr[at]eso[dot]org`

[2]*Joint ALMA Observatory (JAO), Chile*

[3]*National Astronomical Observatory of Japan (NAOJ), Japan*

[4]*National Radio Astronomy Observatory (NRAO), Charlottesville, USA*

Abstract. Astronomers can now find images in the ALMA Science Archive (ASA `https://almascience.org/aq`) that are similar to a given image they are looking at. The web interface allows them to interactively and iteratively refine their selection to match the similarity they are after.

We report here about the progress that has been made since ADASS 2023, where the methodology of the self-supervised contrastive affine-transformation-independent Deep Learning was presented. We describe the features of the implemented production version of the interface that provides similarities for over 500,000 images and give an outlook for possible future improvements.

1. Introduction

The amount of astronomical data is growing exponentially (e.g. Stoehr 2019) resulting in an exponential increase of data available per astronomer. Not data but astronomers are becoming - or have already become - the rare resource in astronomy (Stoehr et al. 2015). For science archives like the ALMA Science Archive (ASA), this evolution means that the next frontier is to make the actual content of the data searchable (Durand et al. 2017), in addition to the searches offered on observational metadata. ALMA has taken small initial steps into this direction (Stoehr et al. 2022): users can search for similar abstracts of publications and projects on the ASA query interface using a state-of-the-art text similarity method. Additionally, we offer preview images that show lines identified through the ALMA Data Mining Toolkit (ADMIT) (Teuben et al. 2015). In Stoehr (2025) we have presented our initial work to develop morphological image similarity search for the ASA by using self-supervised contrastive deep representation learning and have presented a proto-type interface for the user-interaction. In this work, we present the refinements done since as well as the actual implementation into the production interface.

2. Refinements

We have extracted the set of 259,126 continuum images and 196,322 peak-flux images of data-cubes from the ALMA Science Archive and have re-trained the contrastive model (Chen et al. 2020) over 10,000 epochs. Tests had shown that the image similarity could result in similar images found despite the fact that the astronomical objects are vastly different in nature, e.g. a protoplanetary disk and a galactic disk. While visually indeed similar, users would like to see only images from similar scientific categories. As the scientific category for each observation is known from the proposal that asked for the observation we have implemented a filtering layer so that for each image, the list of the most similar 1000 images is constructed only from the scientific category of the image or related categories. To identify which categories are related, we carried out a statistical analysis of the possible combinations of science categories used in each of the published scientific articles making use of ALMA data.

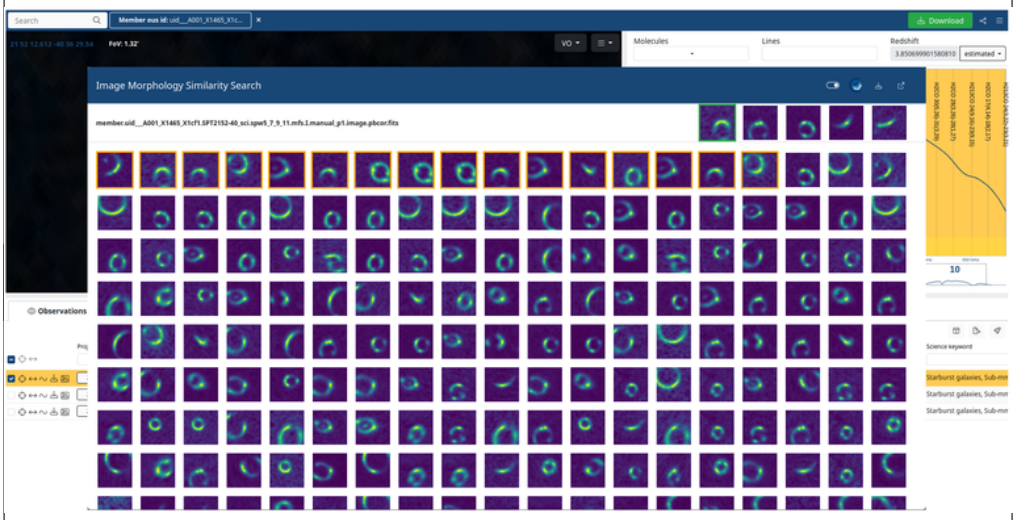


Figure 1. Morphological Similarity Search Interface on the ASA showing the first most-similar images to the original image in the top left corner. Each additional selection on the interface will trigger a reordering of the remaining images so that they are most similar to the combination of all selected images, thus allowing astronomers to interactively refine the search to their needs. Users can send the selected images to CARTA for remote visualisation, obtain a download script for them, or - most importantly - can send the selected observations to a new Archive Query Interface tab where additional subfiltering can be performed.

3. Implementation

For the implementation we used the Angular web framework and followed the prototype design to a large extent. Figure 1 shows the production interface that opens if a user clicks on any of the $\boxed{\sim}$ icons next to the previews in the query interface or on the download page. The most similar images to the selected image (orange border) are shown and more images are available through scrolling. On this interface, the user can now select additional images of interest by clicking with the mouse (or right-click to

select all images up to a given one). Each time they do so, the interface instantaneously updates the display, reordering the remaining images by their simultaneous similarity to all images selected so far. The automatic reordering can be toggled off and on using the control on the right-hand side of the top bar. Hovering over one of the thumbnail images provides additional information about the corresponding observation and FITS file. There are five 'quick-look' images displayed at the top right which correspond to the five largest image groups in the list of 1,000 images computed by using k-means clustering. If the user clicks on one of them, they pre-select similar images in the interface.

After making a selection, users can send all the FITS files corresponding to selected images to the Cube Analysis and Rendering Tool for Astronomy (CARTA) (Wang et al. 2020) with a single click. Users need to follow the instructions at the top of the CARTA pop-up to see all images there. They can also click on the download icon to get a download script for those FITS files.

The main functionality however is to allow users to open a new ASA query tab with the observations corresponding to the selected images through the right-most icon in the top bar. In the new query tab users can further filter the selected observations using all criteria available there or retrieve additional metadata or look at the previews including the spectra. We anticipate that this allows users to find very quickly observations they might be interested in.

To honour the proprietary period, the similarity interface only displays images of public data. As the inference is very fast to compute - a couple of hours for 200,000 images -, we can construct a new similarity matrix for the query interface regularly to include images that have become public recently. The web-interface itself only relies on the pre-computed similarity matrix and makes use of standard web technology. It does not need to be connected to a GPU.

4. Outlook

So far the hyperperparameters are optimised using only visual inspection. However, in many cases we do have several images for each source in a dataset. This allows us to construct an overall score value from the similarity distance of images that we know are from the same source. We then can run a hyperparameter search to fine-tune parameter setting minimising the score value automatically. We are also considering collecting the selections users are making on the ASA interface - entirely anonymously - and using those as further input to improve the hyperparameters in future re-trainings.

Furthermore, it is conceivable to automatically derive labels and source descriptions for the images from the source names and their details given in ALMA publications and proposal abstracts. This information might be then used together with the identified similarity matrix to label images without labels as well as to offer chat-bot functionality on the text, thus allowing users to query images through natural language.

Acknowledgments. ALMA is a partnership of ESO (representing its member states), NSF (USA) and NINS (Japan), together with NRC (Canada), NSTC and ASIAA (Taiwan), and KASI (Republic of Korea), in cooperation with the Republic of Chile. The Joint ALMA Observatory is operated by ESO, AUI/NRAO and NAOJ.

References

Chen, T., Kornblith, S., Norouzi, M., & Hinton, G. 2020, A Simple Framework for Contrastive Learning of Visual Representations. eprint: 2002.05709

Durand, D., Haase, J., Swam, M., Fabbro, S., & Goliath, S. 2017, in Astronomical Data Analysis Software and Systems XXV, edited by N. P. F. Lorente, K. Shortridge, & R. Wayth, vol. 512 of Astronomical Society of the Pacific Conference Series, 351

Stoehr , F. 2025, in Astronomical Society of the Pacific Conference Series, edited by A. Jacques, R. Seaman, N. Gandilo, & T. Linder, vol. 541 of Astronomical Society of the Pacific Conference Series, 173

Stoehr, F. 2019, in Astronomical Data Analysis Software and Systems XXVII, edited by P. J. Teuben, M. W. Pound, B. A. Thomas, & E. M. Warner, vol. 523 of Astronomical Society of the Pacific Conference Series, 387

Stoehr, F., Lacy, M., Leon, S., Muller, E., & Kawamura, A. 2015, in Astronomical Data Analysis Software an Systems XXIV (ADASS XXIV), edited by A. R. Taylor, & E. Rosolowsky, vol. 495 of Astronomical Society of the Pacific Conference Series, 69

Stoehr, F., Manning, A., McLay, S., Ashigatawa, K., del Prado, M., Jenkins, D., Damian, A., Wang, K. S., Moraghan, A., Plunkett, A., Lipnicky, A., Sanhueza, P., Calistro Rivera, G., & Gaudet, S. 2022, The Messenger, 187, 25. eprint: 2208.03275

Teuben, P., Pound, M., Mundy, L., Rauch, K., Friedel, D., Looney, L., Xu, L., & Kern, J. 2015, in Astronomical Data Analysis Software an Systems XXIV (ADASS XXIV), edited by A. R. Taylor, & E. Rosolowsky, vol. 495 of Astronomical Society of the Pacific Conference Series, 305

Wang, K. –., Comrie, A., Harris, P., Moraghan, A., Hsu, S. –., Pinska, A., Chiang, C. –., Jan, H., Simmonds, R., Pang, Q., & et al. 2020, in Astronomical Data Analysis Software and Systems XXIX, edited by R. Pizzo, E. R. Deul, J. D. Mol, J. de Plaa, & H. Verkouter, vol. 527 of Astronomical Society of the Pacific Conference Series, 213

Astronomical Data Analysis Software and Systems XXXV
*ASP Conference Series, Vol. **VOL** Compiled: Tuesday 30th June, 2026@ 01:30*
K. Polsterer, D. Bohlender, S. Gwyn, A. Hincks and P. Teuben, eds.
© **YEAR** Astronomical Society of the Pacific

ZZZDEBUG!204.tex
Boch, T.|textbf
Baumann, M.
Fernique, P.

HiPS2FITS 3D: a new service to generate cube cutouts from HiPS 3D datasets

Thomas Boch,[1] Matthieu Baumann,[1] and Pierre Fernique[1]

[1]*Observatoire astronomique de Strasbourg, Université de Strasbourg, CNRS, UMR 7550, Strasbourg, F-67000, France* thomas.boch@astro.unistra.fr

Abstract. The Hierarchical Progressive Survey (HiPS) standard has become a widely adopted framework for the visualisation and exploration of astronomical data across spatial and now spectral and temporal dimensions (Fernique et al. 2026). While the HiPS2FITS service has long enabled users to extract two-dimensional image cutouts from HiPS image surveys, the growing availability of HiPS 3D datasets (representing spectral or temporal cubes) has created a demand for an equivalent capability in three dimensions.

We present HiPS2FITS 3D, a new service developed at CDS to generate spectral cube cutouts directly from HiPS 3D datasets. This service enables astronomers to retrieve physically meaningful data cube subsets for further analysis in dedicated tools, once a region of interest has been identified using HiPS 3D visualisation clients such as Aladin Lite (Baumann et al. 2026), Aladin Desktop, or GLUE. We describe the design of the service, its API, implementation details, performance measurements, current limitations, and future improvements.

1. Motivation: From HiPS visualisation to quantitative analysis

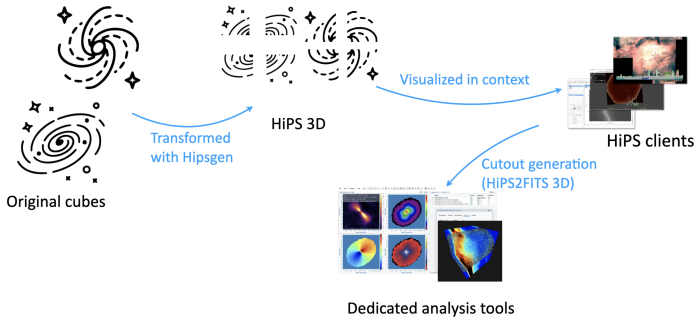


Figure 1. HiPS2FITS 3D in the visualisation in context/analysis workflow

HiPS 3D extends the HiPS concept beyond two-dimensional imagery by organizing spectral cubes into hierarchical multi-resolution structures. This allows interactive visualisation of large radio or spectral-line surveys directly in HiPS-enabled clients. However, visualisation alone is not sufficient for scientific exploitation: astronomers ultimately require local data extraction to perform quantitative analysis using dedicated

software environments.
HiPS2FITS 3D fills this gap by acting as a bridge between visualisation and analysis, as depicted in figure 1. After a user visually identifies a spatial and spectral region of interest within a HiPS 3D client, the service can be queried to retrieve a corresponding cube cutout in standard FITS format (or alternative formats). This workflow mirrors the widely used HiPS2FITS service for 2D images, but now extends the concept into full three-dimensional data access.
This new capability significantly improves data accessibility for large surveys such as Galactic HI cubes, CO surveys, or upcoming large spectral datasets such as SKA (Dewdney et al. 2009). It allows users to remain within a HiPS-based discovery environment while still obtaining scientifically usable data products for offline or remote analysis.

2. API and User Interface

The HiPS2FITS 3D service is deployed as an HTTP-based API. Users can formulate requests by specifying both spatial and spectral parameters in a single query.
Users can access the service at `https://alasky.cds.unistra.fr/hips-cube-services/hips2fits3d`.

Several important features define this API:

HiPS 3D dataset selection

Users may select any HiPS 3D dataset hosted at CDS. A full list of available HiPS 3D surveys is published online at `https://aladin.cds.unistra.fr/java/HipsList3D.txt`

Spatial projection parameters

Spatial query parameters can be provided in two alternative forms:

- A direct WCS dictionary specifying projection metadata.
- A simpler description consisting of a central sky position, requested angular size, and projection type.

Spectral range selection

The spectral axis is defined through lower and upper frequency boundaries, expressed in Hertz. This allows for extraction of contiguous spectral slabs from the HiPS cube.

Output formats

While FITS is the primary scientific output, the service can also return data products in visualisation-oriented formats: MP4 video, animated GIF, PNG mosaic grid.
These options make the service suitable not only for scientific analysis but also for outreach or rapid-look inspection.
This dual approach accommodates both advanced users who wish to control WCS metadata precisely and users who prefer a simplified sky-region definition.

3. Implementation and performance

Software Architecture

As the HiPS2FITS image cutout service, HiPS2FITS 3D is implemented entirely in Python, building upon established software components:

- *cdshealpix* (Rust-based) provides high-performance HEALPix indexing.
- *Astropy* is used for WCS management and FITS handling.
- *Numba* accelerates bilinear resampling operations on the spatial axes.

This combination ensures both correctness in astrometric transformations and efficiency in data access.

Resampling strategy

The service performs bilinear resampling in the spatial dimensions when assembling cutouts from HiPS tiles. Currently, no resampling is performed along the spectral axis. Instead, spectral planes are extracted directly.

Performance measurements

Benchmark tests were run on the GALFA HI HiPS 3D dataset. A representative cutout of size 512×512 pixels and 1300 spectral channels was generated using 16 threads in 120 seconds, corresponding to a throughput of about 3 million pixels per second.

This performance demonstrates the feasibility of on-demand cube generation for moderate cutout sizes and validates the architectural choices of combining Rust-based tile access with Python-level orchestration and Numba acceleration.

4. Current limitations

The HiPS2FITS 3D service currently exhibits several limitations that guide potential ongoing development.

Coupled spatial and spectral resolution

In the present HiPS 3D structure, spatial and spectral axes are bound together. This makes it difficult to generate cutouts with low spatial resolution but high spectral resolution for instance.

No spectral resampling

While spatial bilinear resampling is implemented, the spectral axis is not resampled. Users therefore receive spectral slices at native resolution and sampling only.

In-memory cutout assembly

Large cutouts are currently assembled fully in memory before being written to output. This behaviour limits maximum cutout size and increases memory pressure.

Transfer bottlenecks

For very large cubes, data transfer time becomes significant after the cutout generation.

CDS-hosted dataset restriction

Our current implementation of the service is limited to HiPS 3D datasets hosted at CDS. It cannot yet generate cutouts from remote HiPS 3D nodes.

5. Future improvements and roadmap

Several enhancements are planned to improve the service.

Streaming cutout generation

By exploiting absolute indexing in both spatial and spectral HiPS dimensions, future implementations could stream data directly to the output file instead of building the whole cube in memory. This would improve scalability and reduce memory constraints.

Asynchronous interface

Exposing the service through an asynchronous UWS (Universal Worker Service) interface would allow users to submit long-running cutout jobs and retrieve results later, improving usability for large data requests.

Compression options

Support for RICE compression in FITS outputs will reduce file size and transfer time.

SF-MOC integration

Updating the MOCPy library to support Spectral-Frequency MOCs (SF-MOC) would eliminate the need to guess which HiPS tiles exist in both spatial and spectral dimensions, improving the robustness of cutout generation.

Integration into Astroquery

Ultimately, HiPS2FITS 3D will be accessible through the *astroquery.cds* module, enabling seamless access from Python data analysis workflows.

Conclusion

We have presented HiPS2FITS 3D, a new service enabling the extraction of spectral cube cutouts from HiPS 3D datasets. It extends the HiPS ecosystem from visualisation into full three-dimensional data access, providing a powerful new tool for the analysis of current and upcoming large spectral astronomical surveys.

References

Baumann, M., Fernique, P., & Boch, T. 2026, in ADASS XXXV, edited by TBD (San Francisco: ASP), vol. TBD of ASP Conf. Ser., TBD

Dewdney, P. E., Hall, P. J., Schilizzi, R. T., & Lazio, T. J. L. W. 2009, IEEE Proceedings, 97, 1482

Fernique, P., Allen, M., Baumann, M., Boch, T., Bot, C., & Pineau, F.-X. 2026, in ADASS XXXV, edited by TBD (San Francisco: ASP), vol. TBD of ASP Conf. Ser., TBD

Astronomical Data Analysis Software and Systems XXXV
*ASP Conference Series, Vol. **VOL** Compiled: Tuesday 30th June, 2026@ 01:30*
K. Polsterer, D. Bohlender, S. Gwyn, A. Hincks and P. Teuben, eds.
*© **YEAR** Astronomical Society of the Pacific*

ZZZDEBUG!205.tex
Van-Lane, P.|textbf
Broekgaarden, F.
Wagg, T.
Duarte, J.
Gesing, S.
Katz, D. S.
Williams, P.
Zonca, A.
FOOBAR!conference!ADASS 2019
FOOBAR!organisations!ASP

Standardizing Software Citation in Astronomy

Phil Van-Lane[1,2], Floor Broekgaarden[1], Tom Wagg[3], Javier Duarte[1], Sandra Gesing[4], Daniel S. Katz[5], Peter Williams[6], and Andrea Zonca[4]

[1]*University of California San Diego, San Diego, CA, United States;* pvanlane@ucsd.edu

[2]*University of Toronto, Toronto, ON, Canada*

[3]*Flatiron Institute, New York, NY, United States*

[4]*San Diego Supercomputer Center, San Diego, CA, United States*

[5]*University of Illinois Urbana-Champaign, Urbana, IL, United States*

[6]*Center for Astrophysics, Cambridge, MA, United States*

Abstract. Software development has become an essential part of astronomy and all of its sub-fields. Because of that, software citation is crucial for crediting earlier work, motivating funding, and encouraging reproducible and collaborative science. However, the community lacks a standardized citation practice for astronomy software. This has led to disparate software citation practices that are difficult to reconcile, increasing the risk of uncredited software and results that cannot be reproduced.

We are addressing this challenge by (i) facilitating the standardization of software citation practices throughout the astronomy community, and (ii) further developing a leading class tool to implement these standards: *The Software Citation Station* (Wagg & Broekgaarden 2024; Wagg et al. 2025). To accomplish these, we will be hosting a workshop and hackathon in April 2026, which will bring together software citation experts across the field to align on a definitive set of best practices. These recommendations will help define the tools required by our community, directing the next phase of development for *The Software Citation Station*.

1. Software Citation in Astronomy

Software development is crucial for astronomy research, but is often not adequately supported, as highlighted in the Astro2020 Decadal Survey (National Academies of Sciences, Engineering, and Medicine 2021). In addition to the development itself, software citation practices are an important tool across all of the sciences to properly attribute and credit earlier work, motivate funding, enable peer-review, validate and reproduce findings, support collaboration and community, and encourage reuse and extension of the work of others (Katz et al. 2020).

While a well-developed ecosystem of tools and services exists to manage citations to traditional publications, such as ADS/SciX, this infrastructure struggles for software citation due to heterogeneity in both the types of software artifacts (e.g., source code, packages, versions) that developers expose, and the methods of citing the software (if

any) that researchers use (e.g., Bouquin et al. 2023, 2020; Li et al. 2017). Even when software authors explicitly provide users with a preferred citation mechanism, this is often not followed.

Because software citation mechanisms are so inconsistent and often incomplete, citations are extremely challenging to track. Different software users may use different keywords or references for the same artifacts, and policies for software citation are not consistent across journals. Although homogenizing these practices will take time and effort, astronomy research will benefit greatly from that. This process can also be streamlined by: (i) clearly articulating best practices for the community of software authors and users (and ensuring these are widely distributed and accessible), and (ii) ensuring that there are software and systems available which are designed to drive adoption and make these practices as painless as possible.

2. Defining Best Practices for the Community

To tackle this challenge, we are hosting a four-day software citation workshop and hackathon at UC San Diego in April 2026. This will bring together experts from the software citation community, including expected representation from ADS, ArXiv, ASCL, academic journals, and researchers.

Previously, the FORCE11 Software Citation Working Group (Katz et al. 2020; Martone et al. 2014; Smith et al. 2016) produced guidelines for software citation, and several astronomy journals have developed software policies using these (Timmes & Muench 2019; Vishniac & Lintott 2016). While these resources provide recommendations to address many general challenges, there are some specific issues for astronomy that need to be resolved. These include:

- How to (and when to) cite software dependencies, and how to handle version control and/or multiple DOIs (Sochat 2022).
- How to (and when to) include acknowledgements that are specifically written by the developer.
- How to cite common programming tools in astronomy such as `python`, `R`, and `matlab` that do not have standardized references and contain many software packages.
- Best practices for including citation format files (such as `CITATION.cff` and `CodeMeta.json`) with astronomy software packages.
- Best practices for both developers and users to ensure astronomy research is FAIR (Findable, Accessible, Interoperable, Reusable) (Wilkinson et al. 2016; Patel 2024).

We will address these in the workshop, and collaboratively write a whitepaper for the community and journals outlining best practices for developers to make astronomy software citeable, and for researchers citing astronomy software.

3. The Software Citation Station

One historical gap has been the lack of a user-friendly and community-supported database of software citation information, which is critical to support and advance the use of soft-

ware and software careers in astronomy (e.g., Bouquin et al. 2023, 2020; Druskat et al. 2024, 2021; Katz & Chue Hong 2018; Katz et al. 2020).

This gap has already been partially filled by *The Software Citation Station* (hereafter *SCS*; Wagg & Broekgaarden 2024; Wagg et al. 2025), which provides tooling to streamline the citation process. On this site, researchers need only to select the software used in their work, and *SCS* automatically provides the citation text and all associated references (including dependencies) for the selected astronomy software. This enables researchers to easily include formal standardized citations with a few clicks, using a system that is flexible enough to adhere to the requirements outlined by each software package.

SCS has been well received by the astronomy community. It has already been used and cited in over 100 papers (see ADS citations), received positive feedback in one of the leading software blogs in astronomy (Williams 2024), and is already being recommended to authors by AAS Publishing [1]. It has the potential to become a core resource for software citations in astronomy, and during the workshop we will begin to implement several concrete next steps to help make it a world-leading tool:

- Increasing the support for version control to improve reproducibility of astronomy research (Chen & Davidson 2020).
- Integrating *SCS* with other efforts for unique software identifiers including HERMES (Druskat et al. 2022), CiteAs (Du et al. 2021), ASCL (Nemiroff & Wallin 1999; Allen et al. 2020), and Asclepias (Henneken et al. 2022).
- Moving *SCS* to an independent website (or part of ADS) and making it easier for astronomers to contribute and edit citations.
- Creating definitive instructions and tutorials for adding software to *SCS*.

Additionally, we will use the workshop to identify further enhancements we can make to support and streamline community best practices.

4. Driving Adoption and Impact

By defining best practices for software citation in astronomy, and coupling that with the further development of *SCS* to support those practices, we expect to significantly impact the astronomy community in several interconnected ways.

First, the improved *SCS* together with the guidelines will increase the number of astronomers that cite software and provide standardized citations to their software by simplifying and clarifying the process. This will also help improve the reproducibility of astronomy research. Second, it will lead to more consistent citations, which is important to quantify and measure the importance of software in astronomy and fund it accordingly going forward. Third, it will provide more evidence to acknowledge the increasingly important role software and development plays in astronomy research, and thereby support the careers of software focused astronomers. Last, it will provide a leading example for other science communities for how to support software citations in their fields.

[1] See `https://journals.aas.org/policy-statement-on-software/`

References

Allen, A., Berriman, G. B., DuPrie, K., Mink, J., Nemiroff, R., Ryan, P. W., Schmidt, J., Shamir, L., Shortridge, K., Taylor, M., Teuben, P., Wallin, J., & Warmels, R. H. 2020, in Astronomical Data Analysis Software and Systems XXVII, edited by P. Ballester, J. Ibsen, M. Solar, & K. Shortridge, vol. 522 of Astronomical Society of the Pacific Conference Series, 731

Bouquin, D., Trisovic, A., Bertuch, O., & Colón-Marrero, E. 2023, arXiv e-prints, arXiv:2302.07500. eprint: 2302.07500

Bouquin, D. R., Chivvis, D. A., Henneken, E., Lockhart, K., Muench, A., & Koch, J. 2020, ApJS, 249, 8

Chen, L., & Davidson, S. B. 2020, in 2020 IEEE 36th International Conference on Data Engineering (ICDE), 1754

Druskat, S., Bertuch, O., Juckeland, G., Knodel, O., & Schlauch, T. 2022, arXiv e-prints, arXiv:2201.09015. eprint: 2201.09015

Druskat, S., Chue Hong, N. P., Buzzard, S., Konovalov, O., & Kornek, P. 2024, arXiv e-prints, arXiv:2402.14602. eprint: 2402.14602

Druskat, S., Katz, D. S., & Todorov, I. T. 2021, arXiv e-prints, arXiv:2103.06681. eprint: 2103.06681

Du, C., Cohoon, J., Priem, J., Piwowar, H., Meyer, C., & Howison, J. 2021, in Companion Publication of the 2021 Conference on Computer Supported Cooperative Work and Social Computing (New York, NY, USA: Association for Computing Machinery), CSCW '21 Companion, 218–221. URL https://doi.org/10.1145/3462204.3482889

Henneken, E. A., Blanco-Cuaresma, S., Accomazzi, A., Muench, A., Ioannidis, A., Nielsen, L. H., & Steffen, J. 2022, Bulletin of the AAS, 54. Https://baas.aas.org/pub/2022i046

Katz, D. S., & Chue Hong, N. P. 2018, arXiv e-prints, arXiv:1807.08149. eprint: 1807.08149

Katz, D. S., Hong, N. P. C., Clark, T., Muench, A., Stall, S., Bouquin, D., Cannon, M., Edmunds, S., Faez, T., Feeney, P., et al. 2020, F1000Research, 9

Li, K., Yan, E., & Feng, Y. 2017, Journal of Informetrics, 11, 989. URL https://www.sciencedirect.com/science/article/pii/S1751157717300329

Martone, M., et al. 2014, San Diego CA: Force11, 10

National Academies of Sciences, Engineering, and Medicine 2021, Pathways to Discovery in Astronomy and Astrophysics for the 2020s (Washington, DC: The National Academies Press). URL https://nap.nationalacademies.org/catalog/26141/pathways-to-discovery-in-astronomy-and-astrophysics-for-the-2020s

Nemiroff, R. J., & Wallin, J. F. 1999, in American Astronomical Society Meeting Abstracts #194, vol. 194 of American Astronomical Society Meeting Abstracts, 44.08

Patel, B. 2024, Presentation - Making FAIR Fair to the Researchers (Force11 Conference - Force2024). URL https://doi.org/10.5281/zenodo.13159874

Smith, A. M., Katz, D. S., & Niemeyer, K. E. 2016, PeerJ Computer Science, 2, e86

Sochat, V. 2022, Journal of Open Source Software, 7, 4458

Timmes, F. X., & Muench, A. 2019, in Astronomical Data Analysis Software and Systems XXVII, edited by P. J. Teuben, M. W. Pound, B. A. Thomas, & E. M. Warner, vol. 523 of Astronomical Society of the Pacific Conference Series, 693

Vishniac, E. T., & Lintott, C. 2016, AJ, 151, 21

Wagg, T., Broekgaarden, F., Van-Lane, P., Wu, K., & Gültekin, K. 2025, Tomwagg/software-citation-station: v1.4. URL https://doi.org/10.5281/zenodo.17654855

Wagg, T., & Broekgaarden, F. S. 2024, arXiv e-prints, arXiv:2406.04405. eprint: 2406.04405

Wilkinson, M. D., Dumontier, M., Aalbersberg, I. J., Appleton, G., Axton, M., Baak, A., Blomberg, N., Boiten, J.-W., da Silva Santos, L. B., Bourne, P. E., et al. 2016, Scientific data, 3, 1

Williams, P. K. G. 2024, The software citation station, https://newton.cx/~peter/2024/citation-station/. URL https://newton.cx/~peter/2024/citation-station/

Astronomical Data Analysis Software and Systems XXXV
*ASP Conference Series, Vol. **VOL** Compiled: Tuesday 30th June, 2026@ 01:30*
K. Polsterer, D. Bohlender, S. Gwyn, A. Hincks and P. Teuben, eds.
*© **YEAR** Astronomical Society of the Pacific*

ZZZDEBUG!207.tex
Ren, J.|textbf
Liu, C.
Ao, Y.
Lin, Z.
Ling, C.
Nie, J.
Niu, B.
Peng, X.
Shao, L.
Tan, S.
Wang, R.
Wei, P.
Wei, S.
Xu, C.
Yao, S.
Zhao, B.
Zhou, Z.
observatories!space-based
calibration
surveys
FOOBAR! ascl:1101.010

On-Orbit Calibration System for the Chinese Space Station Survey Telescope

Juanjuan Ren,[1] Chao Liu,[1] Yiping Ao,[4] Zhenhui Lin,[4] Chenxiaoji Ling,[1] Jundan Nie,[1] Bingli Niu,[2] Xiyan Peng,[3] Li Shao,[1] Siyuan Tan,[4] Rui Wang,[1] Peng Wei,[1] Shoulin Wei,[5] Chun Xu,[3] Su Yao,[1] Bowei Zhao,[1] and Zhimin Zhou[1]

[1]*National Astronomical Observatories, Chinese Academy of Sciences, Beijing 100101, China;* jjren@nao.cas.cn

[2]*Nanjing Institute of Astronomical Optics & Technology, Chinese Academy of Sciences, Nanjing 210042, China*

[3]*Shanghai Astronomical Observatory, Chinese Academy of Sciences, Shanghai 200030, China*

[4]*Purple Mountain Observatory, Chinese Academy of Sciences, Nanjing 210023, China*

[5]*Faculty of Information Engineering and Automation, Kunming University of Science and Technology, Kunming 650500, China*

Abstract.
As a forthcoming Stage-IV space-based survey telescope, the Chinese Space Station Survey Telescope (CSST) presents extraordinary calibration challenges. We introduce a novel On-Orbit Calibration System architected around three integrated subsystems: a Dynamic and Self-Adaptive Calibration Subsystem for intelligent and real-time calibration decision-making; a Calibration Product Generation Subsystem for high-throughput processing; and a Calibration Product Management Subsystem ensuring version control and traceability. This closed-loop intelligent calibration ecosystem is designed to guarantee the long-term data stability and accuracy essential for CSST's flagship surveys, while also providing a valuable reference for future missions.

1. Introduction

The Chinese Space Station Survey Telescope (CSST) is a next-generation 2-meter space astronomical facility, uniquely characterized by its high spatial resolution ($R_{\rm EE80} < 0.15''$), a wide field of view (1.72 deg^2), and multi-wavelength coverage from the near-ultraviolet (0.255 μm) to near-infrared (2.5 μm) as well as the Terahertz band (Zhan 2011, 2025; CSST Collaboration et al. 2025). Its innovative off-axis three-mirror anastigmat (TMA) design feeds five scientific instruments: the Main Survey Camera (MSC) for multi-band imaging and slitless spectroscopy survey, the high-resolution Multi-Channel Imager (MCI; Zheng et al. 2025), the Integral Field Spectrograph (IFS), the Cool Planet Imaging Coronagraph (CPI-C), and the Terahertz Spectrometer (TS). Co-orbiting with the China Space Station at an altitude of approximately 400 km, CSST's 10-year mission

will conduct a revolutionary 17,500 deg^2 wide-field survey and 400 deg^2 deep-field survey. The powerful suite of scientific instruments enables CSST to address fundamental questions across modern astronomy (Gong et al. 2025), particularly in precision cosmology.

Leveraging the full scientific value of CSST data requires overcoming significant calibration challenges. The space environment induces gradual changes in detector characteristics and optical performance, while the data volume expected to reach ~5 PB necessitates highly automated processing. To ensure the high quality of its scientific data products, the CSST calibration system must be a dynamic, self-adaptive, and highly automated infrastructure. It is designed to supersede fixed-schedule calibration with an intelligent data-driven approach, thereby ensuring the data quality throughout the mission's decade-long lifespan.

2. Calibration Challenges and Requirements

The unprecedented capabilities of CSST pose significant challenges for its on-orbit calibration. These challenges fall into three interrelated aspects:

- Instrumental Stability and Evolution: Key detector parameters including but not limited to charge transfer inefficiency (CTI), dark current, and flat-field response, evolve over time due to cumulative radiation exposure and thermal fluctuations. A traditional static calibration approach is wholly inadequate; instead, a calibration system capable of continuous monitoring and self-adaptive updates is essential.
- Big Data and Processing Challenges: The CSST calibration observations will generate a vast and high-frequency volume of data. The system must handle this vast data volume with high throughput and low latency to ensure rapid product generation, enabling near-real-time assessment of instrument health and timely feedback for calibration strategy optimization.
- Systemic Integrity and Flexibility: Beyond processing, the lifecycle management of calibration products demands rigorous version control for all data, algorithms, and the rules governing their application. Such traceability is essential for ensuring the reproducibility of any scientific result. Furthermore, the system must be flexible enough to generate, certify and allocate various specialized calibration products, driven by diverse scientific requirements.

3. CSST On-Orbit Calibration System

3.1. System Architecture Overview

The CSST On-Orbit Calibration System is designed as an integrated framework to address the challenges outlined in Section 2. Three core subsystems are integrated into a cohesive and closed-loop workflow. As depicted in Figure 1, the process begins with the Dynamic and Self-Adaptive Subsystem. This subsystem continuously assesses the instrument status using science data and engineering telemetry, and it then generates an optimal calibration strategy, thus triggering calibration observation activities as needed.

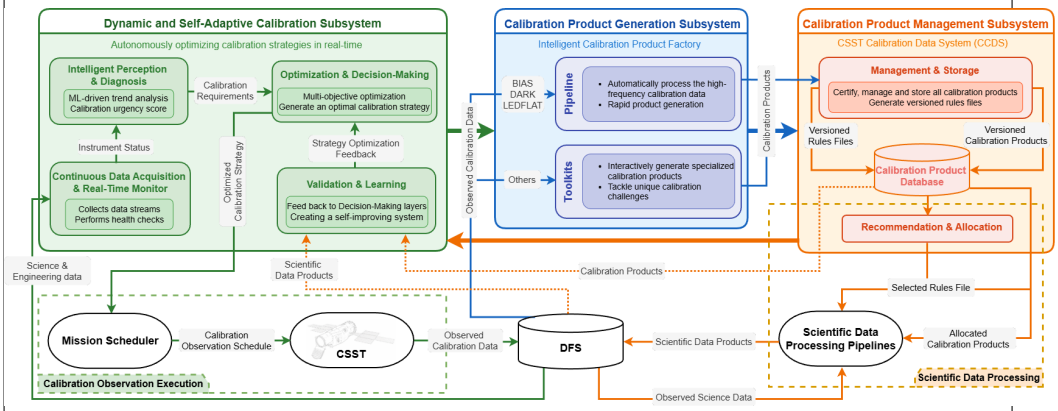


Figure 1. Data flow and system architecture of the CSST On-Orbit Calibration System.

The observed calibration data are then processed by the Calibration Product Generation Subsystem, which features an automated pipeline for routine calibration products and interactive toolkits for specialized analysis. The resulting calibration products are then certified, versioned, and stored by the Calibration Product Management Subsystem. Finally, this management subsystem automatically allocates the appropriate calibration products to the scientific data processing pipelines, thereby guaranteeing full traceability and consistency across the mission.

3.2. Dynamic and Self-Adaptive Calibration Subsystem

A core innovation of the CSST calibration system is the Dynamic and Self-Adaptive Calibration Subsystem, which enables a fundamental shift from a static and schedule-based calibration paradigm to a proactive data-driven one. In contrast to the traditional fixed-interval approach which is inefficient and unable to respond to sudden changes, this subsystem dynamically triggers calibration activities based on the real-time assessment of instrument status. Its primary objective is to dynamically optimize calibration strategies in real time, thus ensuring sustained accuracy while maximizing observational efficiency.

The subsystem executes a continuous five-stage cycle, thus defining a closed-loop and intrinsically self-improving architecture. It begins with the continuous acquisition of science data and engineering telemetry. This data is then ingested by an intelligent perception and diagnosis layer, where machine learning algorithms perform instrument state assessment, forecast trends, and assign a “calibration urgency score”. Based on this diagnosis, an optimization and decision-making layer applies multimodal optimization to derive an optimal calibration strategy. This strategy is then transmitted to the mission scheduler to initiate the required calibration observations. Finally, the resulting data and validation metrics are fed back to refine the diagnostic and decision-making models, thus creating a learning loop that allows the system to adapt throughout the mission’s lifetime.

3.3. Calibration Product Generation Subsystem

Functioning as an intelligent calibration product factory, the Calibration Product Generation Subsystem transforms raw calibration observational data into calibration products. It employs a two-tiered architecture to balance the need for high-throughput automation with expert-driven flexibility. An automated pipeline handles the large-volume high-frequency stream of calibration observational data, to generate the standard and foundational calibration products, thereby ensuring the stability and reproducibility required for survey operations. In parallel, a suite of interactive toolkits enables calibration experts to perform specialized analysis, validate new algorithms, and generate customized calibration products that address complex challenges. This dual-mode design provides the calibration system with both robustness and the flexibility to handle diverse requirements.

3.4. Calibration Product Management Subsystem

The Calibration Product Management Subsystem, also designated the CSST Calibration Data System (CCDS; Zhao et al. 2025), serves as the central repository and data governance layer for the entire calibration ecosystem. It is responsible for the certification, management, and storage of all calibration products. Furthermore, a core function of this subsystem is to generate and maintain versioned rules files, which govern the automated selection and application of appropriate calibration products based on observational parameters. This robust management framework enables another critical function of the subsystem: intelligent recommendation and allocation. Based on the versioned rules files, it automatically allocates the required calibration products to the scientific data reduction pipelines when requested. This automated workflow is essential to ensure data integrity, full traceability, and reproducibility of scientific results.

4. Conclusions

The CSST On-Orbit Calibration System presents a fundamental paradigm shift from a static scheduled approach to an intelligent data-driven framework. By integrating a dynamic decision-making engine, automated calibration product generation, and rigorous data management into a closed-loop architecture, the system fulfills the stringent requirements of a modern large-scale survey mission. This architectural framework offers a scalable reference for future astronomical missions that are expected to handle unprecedented data volumes and calibration complexity.

Acknowledgments. This work is supported by the China Manned Space Project with grant No. CMS-CSST-2025-A19, and the CSST scientific data processing and analysis system of the China Manned Space Project.

References

CSST Collaboration, et al. 2025, arXiv e-prints, arXiv:2507.04618
Gong, Y., et al. 2025, Science China Physics, Mechanics, and Astronomy, 68, 280402
Zhan, H. 2011, Scientia Sinica Physica, Mechanica & Astronomica, 41, 1441
— 2025, Chinese Science Bulletin, 66, 1290
Zhao, B., Wei, S., Ren, J., Wang, F., Ling, C., & Liu, C. 2025, Progress in Astronomy, 43, 354
Zheng, Z.-Y., et al. 2025, arXiv e-prints, arXiv:2509.14691

Astronomical Data Analysis Software and Systems XXXV
*ASP Conference Series, Vol. **VOL** Compiled: Tuesday 30th June, 2026@ 01:30*
K. Polsterer, D. Bohlender, S. Gwyn, A. Hincks and P. Teuben, eds.
*© **YEAR** Astronomical Society of the Pacific*

ZZZDEBUG!208.tex
Schweder, F.|textbf
Polsterer, K. L.
Trujillo-Gomez, S.
algorithm!machine learning!deep learning
methods!Gaussian distributions
astronomy!simulation
FOOBAR! ascl:1101.010

Taking a Look through the Incliscope – A Deep Convolutional Mixed Density Network to predict Galactic Inclinations

Fenja Schweder,[1,2] Kai L. Polsterer,[2] and Sebastian Trujillo-Gomez[2]

[1]*University of Bremen, Bremen, Germany;* fen_sch@uni-bremen.de

[2]*HITS gGmbH, Heidelberg, Germany*

Abstract. Gaining knowledge about the orientation of a disk galaxy allows researchers to draw valuable conclusions about properties like distance, shape, or luminosity. The fact that we are bound to a specific observation point makes it extremely difficult to extract three-dimensional data from observed phenomena.

Determining a galaxy's inclination from an optical image relies mainly on fitting ellipsoids to the disk and solving the intrinsic axis ratio as defined by Hubble and Holmberg. This method is limited by not addressing natural irregularities in the disk shape and neglecting the degeneracy between 3D shape and inclination.

We present the probabilistic framework *Incliscope*, a novel way to estimate inclinations of galaxies based on optical images. Instead of fitting ellipsoids, we use a probabilistic approach to collect properly-calibrated posterior distributions among inclinations from simulated galaxies and use them as training data for a Deep Convolutional Mixture Density Network (DCMDN). The distributions predicted by the trained network reach an average CRPS of $\approx 3°$ compared to the respective true inclination angles. Our method outperforms the isophote-fitting technique in measures of accuracy, stability, and time-efficiency.

1. Introduction

Although modern surveys and observation techniques provide us with rich data among various aspects of our universe, obtaining three-dimensional spatial features of distant celestial objects remains a challenging task. Whenever an observed feature is integrated to perceive two-dimensional image representations, we lose spatial information along the view axis. The lack of this information complicates the derivation of additional properties. We take the observation of nearby galaxies as a generic case. To make assumptions about properties such as shape, brightness, or distance, knowledge of a galaxy's orientation is crucial (Tully & Fisher 1977; van der Kruit & Freeman 2011). Due to the degeneracy that comes with integration along the view axis, it is almost impossible to reconstruct the orientation with sufficient precision (Yip et al. 2011).

In contrast, artificial galaxies from cosmological simulations like Illustris TNG (Nelson et al. 2015) provide the instrinsic shape in all spatial dimensions as a ground truth. By using a simulation-based inference (SBI) approach (Cranmer et al. 2020) to learn likely distributions of spatial information from this ground truth, we are able to fill the gaps for our observed data. In this work, we introduce *Incliscope*, a probabilistic neural network to predict the inclination of a galaxy based on a given optical image.

2. The Incliscope Framework

Our implementation was written in Python and can be orchestrated with YAML configuration files. It consists of the model implementation itself, routines for training and testing a model, as well as scripts to acquire and preprocess the training data, and visualize results.

2.1. Model and Architecture

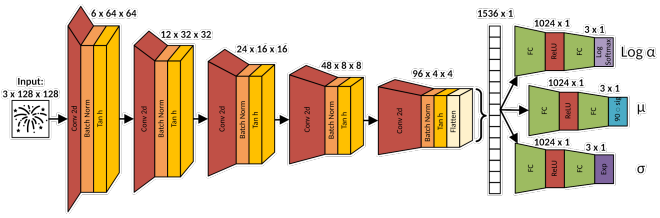


Figure 1. *Incliscope's* architecture with three Gaussian mixture components. Over five convolutional layers, the input image is compressed to a 1D feature vector for which the parameters of a mixed denisty distribution are predicted.

To address the ambiguity of the inclination derived from an optical image, we consider the inclination a probability distribution rather than a scalar value. A Gaussian mixture model as described by McLachlan & Basford (1988) is capable of describing arbitrary probability distributions. Considering a Gaussian distribution

$$\mathcal{N}(x|\mu,\sigma) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{\frac{1}{2}(\frac{x-\mu}{\sigma})^2}, \tag{1}$$

the probability density Φ of a Gaussian mixture model with m components for a random variable x is defined as

$$\mathcal{M}(x \mid \boldsymbol{\alpha},\boldsymbol{\mu},\boldsymbol{\sigma}) = \sum_{i=1}^{m} \alpha_i \cdot \mathcal{N}(x|\mu_i,\sigma_i). \tag{2}$$

In specific, the model consists of m Gaussian distributions that are defined by the means $\boldsymbol{\mu} = \{\mu_1, ..., \mu_m\}$ and the standard deviations $\boldsymbol{\sigma} = \{\sigma_1, ..., \sigma_m\}$. The distributions are weighted by the mixture coefficients $\boldsymbol{\alpha} = \{\alpha_1, ..., \alpha_m\}$.

Instead of predicting the inclination itself, our neural network is required to predict the parameters defining the according distribution, considering the inclination as a random variable. For that purpose, we use a *Deep Convolutional Mixture Density Network (DCMDN)* (D'Isanto & Polsterer 2018) as depicted in Fig. 1. This architecture combines the compression of a 3-channel input image to a 1D feature vector similar to *AlexNet* (Krizhevsky et al. 2017) with the Mixed Density Network (MDN) by Bishop (1994).

2.2. Training

We performed supervised training on the neural network where we minimized the loss between predicted and true inclination distributions via gradient descent. The model was trained for 200 epochs with the ADAM optimization algorithm (Kingma & Ba 2014) and a learning rate $\lambda = 10^{-4}$.

2.2.1. Data

The data to train and test the model originates from the IllustrisTNG simulations (Weinberger et al. 2017; Pillepich et al. 2018). Over 2,500 galaxies consisting of at least 3000 stellar particles were selected. We rendered 33 RGB images from each galaxy by randomly changing the orientation of the particle cloud with respect to the view point, bin the luminosities all particles in the i-, r-, and g-bands and transform them into three-channel images with the algorithm of Lupton et al. (2004). The ground-truth inclination for the galaxy in each image is obtained by calculating the total angular momentum of the respective particles. To receive a ground truth distribution describing the inclination, we used bootstrapping across 1024 randomly assembled subsamples with 70% percent of the particles.

2.2.2. Loss Function

To ensure properly calibrated distributions that are neither over- nor underconfident, the loss between true and predicted inclination is estimated with the *Continuous Ranked Probabilty Score (CRPS)* (Gneiting & Raftery 2007). It is defined as the integral over the difference between the cumulative distribution function and the unit step function at a given observation. For a Gaussian mixture model, Grimit et al. (2006) derived the CRPS as

$$CRPS_{MDN}(\mathcal{M}(x\,|\,\boldsymbol{\alpha},\boldsymbol{\mu},\boldsymbol{\sigma})) = \sum_{i=1}^{m} \alpha_i A(x-\mu_i,\sigma^2) - \frac{1}{2}\sum_{i=1}^{m}\sum_{j=1}^{m} \alpha_i\alpha_j A(\mu_i-\mu_j,\sigma_i^2+\sigma_j^2), \quad (3)$$

where

$$A(\mu,\sigma^2) \equiv 2\sigma\varphi_0\left(\frac{\mu}{\sqrt{\sigma^2}}\right) + \mu\left[2\psi_0\left(\frac{\mu}{\sqrt{\sigma^2}}\right) - 1\right]. \quad (4)$$

Here, φ_0 describes the standard probability density function with a mean of 0 and a standard deviation of 1, while ψ_0 means the standard cumulative distribution function.

3. Performance Evaluation

After running the training as described above, we receive a DCMDN model that can predict mixed density distributions of likely inclination angles for a given set of optical images in less than 0.001 seconds. We evaluate the model by comparing the losses between the predictions and ground truth of a distinct test set to the losses between inclination estimates with an isophote fitting approach and the ground truth.

3.1. Test Scenario

Similar to the assembly of the training set, we collected $\sim$ 400 galaxies from IllustrisTNG that were distinct from the galaxies in the training set. We rendered 9 RGB images per galaxy. Roughly half of the galaxies have a disk morphology. We separated between disk and non-disk galaxies and thus created two test sets. For each test image, we calculated in alternate guess for the inclination by fitting multiple isophotes with the algorithm of Jedrzejewski (1987) to the galaxy shape and solving the axis-ratio-equation (Hubble 1926). With this approach, we receive up to 50 possible ellipse fits, from which we can derive as many inclinations. Treating the inclinations as random

observations, we are able to obtain a Gaussian distribution defined by a mean and a standard deviation, comparable to the predictions of our DCMDN model. Obtaining a normal distribution of inclination angles with this method takes 5 seconds on average per image.

3.2. Results

	Incliscope	**Isophote Fitting**
Disk Galaxies	2.74°	5.69°
Non-Disk Galaxies	7.56°	13.67°

Table 1. The average CRPS loss between predicted distributions and ground truth inclinations.

The average CRPS losses for the predictions by Incliscope and for the isophote fitting estimates measured over all test samples are listed in table 3.2. We see that for disk galaxies, the predicted distributions by Incliscope achieve slightly better CRP scores compared to the ground truth, than the distributions obtained from isophote fitting. For the non-disk galaxies the improvement of the Incliscope predictions is even more significant, although we denote here that isophote fitting is not established on non-disk galaxies.

4. Conclusion and Outlook

The introduced framework *Incliscope* is capable of predicting the inclination for optical images of galaxies by expressing the inclination as a properly calibrated mixed density distribution. This method outperforms the traditional isophote fitting by predicting results closer to the ground truth, predicting better calibrated probability distributions, and being significantly faster and scalable for big data sets. We aim to improve the framework by adding more complex network architectures, performing hyperparameter optimiziations, and setting up a preprocessing routine that allows the transfer from simulated images to real observations.

Acknowledgments. The authors gratefully acknowledge the generous and invaluable support of the Klaus Tschira Foundation.

References

Bishop, C. 1994, Mixture density networks, Workingpaper, Aston University

Cranmer, K., Brehmer, J., & Louppe, G. 2020, Proceedings of the National Academy of Sciences, 117, 30055–30062. URL http://dx.doi.org/10.1073/pnas.1912789117

D'Isanto, A., & Polsterer, K. 2018, A&A, 609, A111. eprint: 1706.02467

Gneiting, T., & Raftery, A. E. 2007, Journal of the American Statistical Association, 102, 359. eprint: https://doi.org/10.1198/016214506000001437, URL https://doi.org/10.1198/016214506000001437

Grimit, E. P., Gneiting, T., Berrocal, V. J., & Johnson, N. A. 2006, Quarterly Journal of the Royal Meteorological Society, 132, 2925. eprint: https://rmets.onlinelibrary.wiley.com/doi/pdf/10.1256/qj.05.235, URL https://rmets.onlinelibrary.wiley.com/doi/abs/10.1256/qj.05.235

Hubble, E. 1926, The Astrophysical Journal, 64, 321

Jedrzejewski, R. I. 1987, MNRAS, 226, 747

Kingma, D. P., & Ba, J. 2014, CoRR, abs/1412.6980. URL https://api.semanticscholar.org/CorpusID:6628106

Krizhevsky, A., Sutskever, I., & Hinton, G. E. 2017, Commun. ACM, 60, 84–90. URL https://doi.org/10.1145/3065386

Lupton, R., Blanton, M. R., Fekete, G., Hogg, D. W., O'Mullane, W., Szalay, A., & Wherry, N. 2004, PASP, 116, 133. eprint: astro-ph/0312483

McLachlan, G. J., & Basford, K. E. 1988, Mixture models: Inference and applications to clustering, vol. 38 (M. Dekker New York)

Nelson, D., Pillepich, A., Genel, S., Vogelsberger, M., Springel, V., Torrey, P., Rodriguez-Gomez, V., Sijacki, D., Snyder, G., Griffen, B., Marinacci, F., Blecha, L., Sales, L., Xu, D., & Hernquist, L. 2015, Astronomy and Computing, 13, 12–37. URL http://dx.doi.org/10.1016/j.ascom.2015.09.003

Pillepich, A., Springel, V., Nelson, D., Genel, S., Naiman, J., Pakmor, R., Hernquist, L., Torrey, P., Vogelsberger, M., Weinberger, R., & Marinacci, F. 2018, MNRAS, 473, 4077. eprint: 1703.02970

Tully, R., & Fisher, J. 1977, A&A, 54, 661

van der Kruit, P., & Freeman, K. 2011, Annual Review of Astronomy and Astrophysics, 49, 301

Weinberger, R., Springel, V., Hernquist, L., Pillepich, A., Marinacci, F., Pakmor, R., Nelson, D., Genel, S., Vogelsberger, M., Naiman, J., & Torrey, P. 2017, MNRAS, 465, 3291. eprint: 1607.03486

Yip, C.-W., Szalay, A., Carliles, S., & Budavári, T. 2011, The Astrophysical Journal, 730, 54. eprint: 1101.5651

Astronomical Data Analysis Software and Systems XXXV
*ASP Conference Series, Vol. **VOL** Compiled: Tuesday 30th June, 2026@ 01:30*
K. Polsterer, D. Bohlender, S. Gwyn, A. Hincks and P. Teuben, eds.
*© **YEAR** Astronomical Society of the Pacific*

ZZZDEBUG!209.tex
Schulz, B.|textbf
Bryant, A.
Früh, J.
Greiner, B.
Hütwohl, M.
Iserlohe, C.
Keilig, T.
Knieling, B.
Krabbe, A.
Schindler, K.
Wiedemann, M.
Zeile, O.
Earth!atmosphere!absorptio
European Virtual Observatory
German Astrophysical Virtual Observatory (GAVO)
Virtual Observatory (VO)
archives!individual!SOFIA
archives!multiple!Infrared Science Archive (IRSA)
computer languages!Python
data!pipelines!processing
data!pipelines!reduction
data!pipelines!science
data!preservation
databases!Postgres
images!processing!source extraction
observatories!ground-based
DaCHS! ascl:1804.005

The SOFIA Data Center (SDC)

Bernhard Schulz,[1] Aaron Bryant,[1] Jonas Früh,[1] Benjamin Greiner,[1] Michael Hütwohl,[1] Christof Iserlohe,[1] Thomas Keilig,[1] Bastian Knieling,[1] Alfred Krabbe,[1] Karsten Schindler,[1] Manuel Wiedemann,[1] and Oliver Zeile[1]

[1]*SOFIA Data Center, Institute for Space Systems, University of Stuttgart, Germany;* bernhard.schulz@irs.uni-stuttgart.de

Abstract. The Stratospheric Observatory for Infrared Astronomy (SOFIA) was a joint project of DLR and NASA, operating from 2010 until 2022 and collecting a wealth of Mid- and Far-Infrared data. This article provides the motivation and an overview of the activities of the SOFIA Data Center (SDC) with some aspects being detailed in additional presentations at this conference. We address the pipeline data reprocessing, the components that add further value to the data products, both the engineering and the VO-compatible archives, science support and the continued development of the pipeline software. Once the 5-year operational phase of the SDC is complete, the archive will be transferred to the new German Astrophysics Center (DZA) as a long-term home.

1. Motivation

With its beginnings in the early 60s, the young field of infrared astronomy went through an exciting time. With a number of very successful stratospheric and space-based observatories like the Convair 990, the KAO, IRAS, ISO, Spitzer and Herschel, the infrared-absorbing effects of the telluric atmosphere were overcome and led to many exiting discoveries. However, the premature end of the airborne observatory SOFIA (Young et al. 2012) in 2022, after less than half of its design lifetime of 20 years, leaves the field with an uncertain future. SOFIA was the last regularly operating observatory sensitive to the mid- and far-infrared (MIR, FIR). The earliest realistic chance for a FIR space observatory is the NASA Probe Explorer mission. There is, however, only a 50 % chance of success, as it is currently also competed for by an X-ray project, rendering the FIR data we already have the more precious.

SOFIA collected about 141 TB of raw data during her operational lifetime. The data comprises the instrument science data, instrument and observatory housekeeping data and all images taken in parallel by the three telescope guider cameras. After observing flights ceased, NASA allowed only one more year for data reprocessing. These activities were limited to science data of Cycles 5 to 9 only.

2. Aims and General Tasks of the SDC

To ensure a thorough scientific exploitation of the data and to create a powerful scientific resource for the time after the mission, the SOFIA Data Center (SDC) was set up in 2024 at the Institute for Space Systems at the University of Stuttgart with a generous

grant by DLR. The SDC aims to bridge the expected 10-20 year gap in availability of regular new FIR data by building a high quality, VO-compatible public SOFIA archive, hosting reprocessed data and up-to-date documentation. This will secure scientific productivity of the SOFIA data for the long term, thus enabling future scientific data exploitation without the need for expert knowledge of SOFIA subtleties. The project is time limited to 5 years with a subsequent transfer of the archive to the DZA in Görlitz. Another important aspect of the plan is to stay in sync with the current SOFIA data repository within IRSA at IPAC/Caltech.

Reprocessing of almost all instrument data,[1] including Cycles 0 to 4 requires creation of an adequate IT-infrastructure. Metadata will have to be reviewed and updated if needed, especially for the earlier observing cycles (See Figure 1 for a schematic). The system supports an iterative approach to the implementation of additional pipeline improvements. Raw data as well as pipeline products will be hosted publicly in a searchable science archive where data will be organized according to astrophysical criteria rather than operational ones. In addition, the SDC is building an Operations- and Engineering-Archive to safeguard essential documentation about the remarkable technological achievement that SOFIA represents, supporting engineering research by interested parties. Astrophysicists will be further supported by updated documentation, development of supporting analysis tools and through webinars, conferences and direct advice.

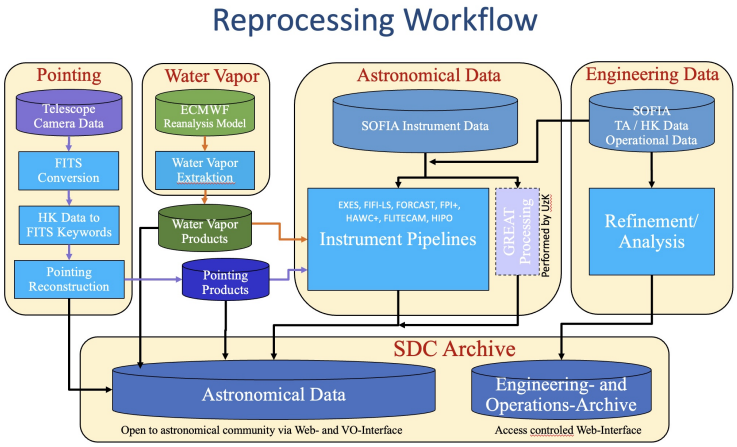


Figure 1. Schematic of the general workflow of the SDC automatic reprocessing system and its components.

2.1. Telescope Pointing Reconstruction

The three CCD cameras WFI, FFI, and FPI+ (Pfüller et al. 2018; Wiedemann et al. 2012) of the telescope assembly (TA) recorded images through all science flights. The SDC, staffed with former German SOFIA Institute (DSI) experts, is in a particularly good position to make this data available. For their first public release, the images will be converted from a native format into astronomer-friendly FITS files with new infor-

[1] GREAT data will be processed in Cologne by the GREAT instrument team.

mative headers, enabling serendipitous science once available in the VO-compatible SDC-archive. In addition the SDC will use these files to generate detailed telescope pointing reconstruction products for every flight, using plate solving techniques. The resulting improved pointing accuracy is particularly beneficial to EXES high resolution spectroscopy and any reconstruction of combined raster-maps.

2.2. Atmospheric Water Vapor Correction

A new method to calibrate the atmospheric transmission, mainly due to its water vapor content, is based on a global atmospheric reanalysis model provided by the European Centre for Medium-Range Weather Forecasts (ECMWF). The method was validated using dedicated, in-flight FIFI-LS observations. It can provide corrections for all data from all SOFIA flights, even after the fact (Fischer et al. 2021; Iserlohe et al. 2021, 2022; Vacca et al. 2023). In particular, data from instruments with low to intermediate spectral resolution will benefit from an implementation of this method into their corresponding processing pipelines.

2.3. The Redux Pipeline Software

The data pipelines for the different science instruments are organized under an umbrella called Redux (Clarke et al. 2015) that provides a homogeneous user interface, requiring only two inputs, a file list and a configuration file. This package is essential for both, the general scientist doing research and requiring an optimally reduced dataset, and the SDC in its quest to re-reduce the entire SOFIA dataset with various improvements.

The Redux development is being continued at the SDC within a GitHub Enterprise Instance at the University of Stuttgart including automatic unit testing[2]. Contributions via issue reports and pull requests on the public GitHub.com repository are welcome and will be synced to the internal server. Guest accounts for outside collaborators are available upon request. Version releases will update the existing PyPI package sofia-redux. More details are provided by Greiner et al. (2026) in this volume.

3. VO-Archive and Data Reprocessing

The SDC Archive is planned to be VO compatible to enable a broad exposure of the data to science searches and minimize the need for new software development through use of available VO-client and server software. DaCHS by Demleitner (2014) will serve to create the VO-archive and provide an API to the internet. The Firefly software suite by Roby et al. (2013) will be used to access this APl as a VO-Client and create the archive webpages.

The SDC will build an automatic reprocessing system that can fully process a set of related observations into a final higher-level product without user interaction. It will be using the instrument pipelines originally constructed by the SMO as well as additional new modules that integrate improved pointing information, corrections for water vapor and other improvements into the processing. Bryant et al. (2026) describe improvements that are already being implemented for the FIFI-LS instrument. The necessary pipeline parameters will be stored in the database for each group and can be

[2]Publicly distributed at `https://github.com/SOFIA-Data-Center/sofia_redux`, which was forked from: `https://github.com/SOFIA-USRA/sofia_redux` and pushes from the internal repository.

refined as the understanding of the data improves. The grouping of related observations will be driven by scientific considerations. Groupings that arose only from operational constraints will be eliminated.

4. Conclusion

The SOFIA Data Center (SDC) at the University of Stuttgart will provide adequate post-processing and a concluding archival phase for SOFIA. The active phase for the SDC will last 5 years. Within this period the data will be substantially improved, placed into a well searchable and documented online-archive. Scientists will receive expert support in understanding residual instrumental effects and finding valuable, yet unpublished data. At the end of this period the archive contents are planned to be transferred into a permanent archive at the DZA.

Acknowledgments. We would like to thank Dr. Markus Demleitner for his support with DaCHS and the conference organizers for providing us with an opportunity to present our project.

References

Bryant, A., Knieling, B., Iserlohe, C., & Früh, J. 2026, in ADASS XXXV, edited by K. Polsterer, & S. Wagner (San Francisco: ASP), vol. TBD of ASP Conf. Ser., 999 TBD

Clarke, M., Vacca, W. D., & Shuping, R. Y. 2015, in Astronomical Data Analysis Software an Systems XXIV (ADASS XXIV), edited by A. R. Taylor, & E. Rosolowsky, vol. 495 of Astronomical Society of the Pacific Conference Series, 355

Demleitner, M. 2014, in Astronomical Data Analysis Software and Systems XXIII, edited by N. Manset, & P. Forshay, vol. 485 of Astronomical Society of the Pacific Conference Series, 309

Fischer, C., Iserlohe, C., Vacca, W., Fadda, D., Colditz, S., Fischer, N., & Krabbe, A. 2021, PASP, 133, 055001. eprint: 2103.16616

Greiner, B., Bryant, A., Früh, J., Hütwohl, M., Iserlohe, C., Keilig, T., Knieling, B., Krabbe, A., Schindler, K., Schulz, B., Wiedemann, M., & Zeile, O. 2026, in ADASS XXXV, edited by K. Polsterer, & S. Wagner (San Francisco: ASP), vol. TBD of ASP Conf. Ser., 999 TBD

Iserlohe, C., Fischer, C., Vacca, W. D., Fischer, N., Colditz, S., & Krabbe, A. 2021, PASP, 133, 055002

Iserlohe, C., Vacca, W. D., Fischer, C., Fischer, N., & Krabbe, A. 2022, PASP, 134, 085001

Pfüller, E., Wolf, J., & Wiedemann, M. 2018, Journal of Astronomical Instrumentation, 7, 1840006

Roby, W., Wu, X., Ly, L., & Goldina, T. 2013, in Astronomical Data Analysis Software and Systems XXII, edited by D. N. Friedel, vol. 475 of Astronomical Society of the Pacific Conference Series, 315

Vacca, W. D., Iserlohe, C., Shenoy, S., Clarke, M., Fischer, C., Thorpe, A., & Chambers, E. 2023, PASP, 135, 085001

Wiedemann, M., Wolf, J., & Roeser, H.-P. 2012, in Ground-based and Airborne Telescopes IV, edited by L. M. Stepp, R. Gilmozzi, & H. J. Hall, vol. 8444 of Society of Photo-Optical Instrumentation Engineers (SPIE) Conference Series, 84442T

Young, E. T., et al. 2012, ApJ, 749, L17. eprint: 1205.0791

Astronomical Data Analysis Software and Systems XXXV
*ASP Conference Series, Vol. **VOL** Compiled: Tuesday 30th June, 2026@ 01:30*
K. Polsterer, D. Bohlender, S. Gwyn, A. Hincks and P. Teuben, eds.
*© **YEAR** Astronomical Society of the Pacific*

ZZZDEBUG!210.tex
Modigliani, A.|textbf
FOOBAR!conference!ADASS 2025
FOOBAR! ascl:1101.010

25 Years working on ESO Automatic Data Reduction Pipelines: Lessons Learned

Andrea Modigliani[1]

Abstract. This contribution reflects on 25 years of experience developing and maintaining automatic data reduction pipelines and related software at the European Southern Observatory (ESO). Over this period, the technology landscape has evolved from early MIDAS-based tools to systems built on the Common Pipeline Library (CPL) and the High-level Data Reduction Library (HDRL), and more recently toward Python-based frameworks such as PyCPL and PyHDRL. Originally designed and implemented as quick-look tools to support Very Large Telescope (VLT) operations, the pipelines later evolved to address the broader needs of the user community by generating science-ready data products. This evolution naturally involved a larger group of stakeholders in verification, validation, and further development. Despite changes in technology and an improved review process using multiple milestones and the involvement of multiple stakeholders, many core challenges and design principles have remained consistent throughout the projects' life cycles, from establishing close collaboration with consortia delivering initial versions, to continuous cooperation with all stakeholders through design, implementation, commissioning, and long-term operations. Key lessons concern the importance of full commitment and valuing complementary contributions from all participants, identifying and prioritizing areas for improvement, and adopting modular, reusable, and sustainable software architectures with well-defined interfaces between scientific and engineering teams. This poster summarizes these lessons and how they have shaped the evolution of ESO's pipeline ecosystem, offering guidance for future projects in astronomical data reduction and analysis.

.

1. Introduction

Operations at the Very Large Telescope (VLT) began in May 1998. Early data reduction pipelines were designed as quick-look tools: robust, fast, and capable of verifying both data quality and instrument health. Over 25 years, pipelines have evolved from MIDAS-based systems to a modern, modular framework using CPL, HDRL, integrating lessons learned from both operational experience and consortium collaborations. Recently ESO released PyCPL and is preparing a first release of PyHDRL.

This paper highlights ESO's pipelines evolution, the critical importance of proper consortium deliveries, and the role of systematic verification at project milestones in ensuring robust, maintainable, and scientifically reliable products.

2. Early Pipeline Development

The first generation of pipelines relied on MIDAS, ESO's in-house software from the 1980s. The Data Flow System (DFS) used the Data Organiser (DO) to classify and

organise raw data, and associate master and static calibrations, and pipeline recipes, and the Reduction Block Scheduler (RBS), to control recipe execution. Initial pipelines were intended primarily for quality control of the data and observations: confirm data quality, avoid reacquisition, and monitor instrument health. Most pipelines were designed for robustness and speed and some could also generate science-ready data products. The FLAMES-UVES pipeline implemented in MIDAS a novel extraction algorithm using the cross-fibre profile obtained from daily calibrations and produced high-quality science data from first light (see Figure 1, left panel, Modigliani et al. (2004)).

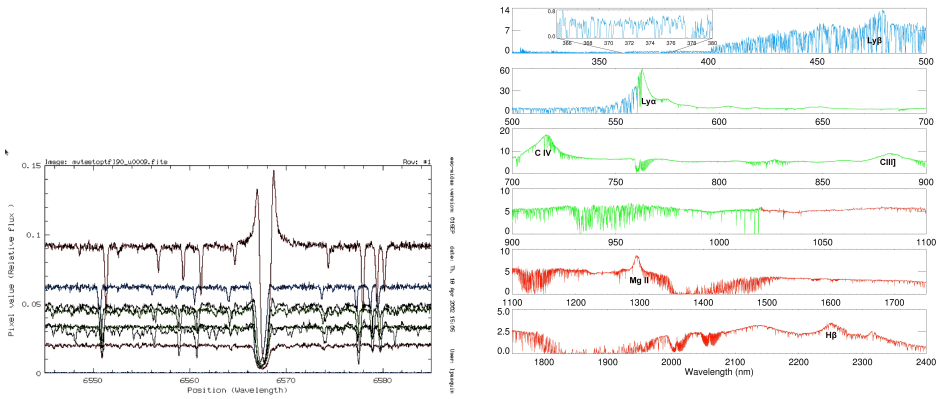


Figure 1. *Left:* The optimally extracted normalised FLAMES 1st light spectrum of Ω_Cen near H_α reduced by the FLAMES-UVES pipeline. *Right:* Pipeline B1422+231 quasar spectrum observed by X-shooter, Vernet et al. (2011).

3. Consolidation and Modernization

To standardize pipeline code, promote reuse, and simplify maintenance and portability, ESO created the Common Pipeline Library (CPL), Banse et al. (2004). The residual MIDAS pipelines were ported to CPL. Among those, the FLAMES-UVES pipeline has been used in the GAIA-ESO survey programme, Sacco et al. (2014).

Then the emphasis for pipeline development shifted to creation of science quality data products. To further standardize, reduce maintenance, simplify desktop users pipeline execution and installation, allow user access to archived science quality pipeline data products, ESO:

- implemented the High-level Data Reduction Library: to share common algorithms across pipelines, Ballester et al. (2015);
- developed ESOReflex: a workflow environment to simplify desktop reductions, Freudling et al. (2013);
- generated RPMs, macports, and introduced Homebrew based installations;
- uses Jenkins as continuous software integration tool;
- defined the Phase 3: formalized delivery standards for science data products (`https://www.eso.org/sci/observing/phase3.html`).

Successful examples include the porting of UVES pipeline to CPL, improving data product quality, and X-shooter pipeline products review, which resulted in a major improvement of the quality of the final science extracted and merged spectrum thanks to a constructive collaboration between ESO staff (see Figure 1, right panel). Figure 2 shows two results obtained with the HDRL based resampling obtained by ERIS-NIX (left) and HAWKI (right).

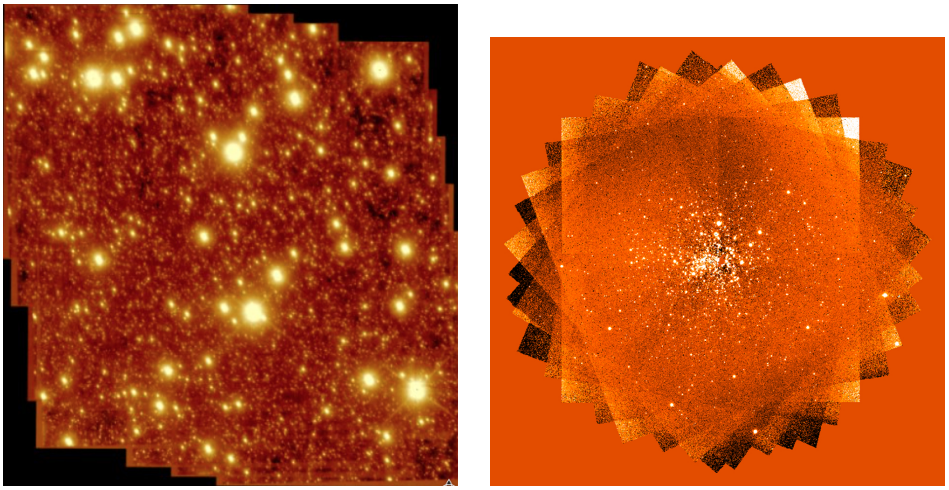


Figure 2. Tiles generated using the HDRL drizzle-resampling function: *left*, eight ERIS-NIX images of Ω_Cen; *right*, 80 HAWK-I flexure frames.

4. Unified Frameworks and Python Integration

Recent ESO developments aim to unify data reduction workflows used in operations and ease consortium contributions and user's installations:

- EDPS/QCFlow: integrates Operations, Quality Control (QC) and desktop-user workflows;
- PyCPL/PyHDRL: Python bindings to access CPL/HDRL objects and functions;
- Homebrew & Apptainer extended pipeline release support to more than 90% of user's common platforms.

5. Lessons Learned

Experience over 25 years shows that pipeline quality depends on careful preparation and verification at every milestone: Preliminary Design Review, Final Design Review, Preliminary Acceptance Europe, commissioning, and start of operations. Problems found late usually trace back to neglected verification or insufficient review early in the process. Key factors for success are:

- strong collaboration between consortia and ESO staff;

- to collect inputs from all stakeholders, including software engineering, algorithm validation, QC, and operations;
- early problem detection and prioritization before milestone reviews;
- robust and maintainable code architecture, allowing multi-platform deployment;
- to focus on both robustness and science quality.

We noted that consortia that implement novel and complex algorithms often are highly motivated to deliver better quality pipelines. Internal collaboration across ESO teams has proven essential for sharing workload, improving quality, and enhancing both stakeholders and user satisfaction.

6. Summary

ESO's DRS evolution-from MIDAS quick-look tools to CPL/HDRL common C libraries, and in the near future the corresponding Python bindings-demonstrates that technical solutions alone are insufficient: success relies on good algorithms, proper delivery, milestone verification, and effective collaboration. Lessons for the next generation of pipelines include: systematic verification at each milestone to prevent late-stage issues; maintaining robust, well refactored, and maintainable software.

These principles ensure pipelines are robust, maintainable, and capable of delivering science-grade products to both operations and desktop users.

Acknowledgments. The author thanks all ESO's stakeholders, the pipeline, science data quality, and Paranal Operations teams, and all collaborating consortia for their contributions to pipeline development and quality improvements.

References

Ballester, P., Gabasch, A., Jung, Y., Modigliani, A., Taylor, J., Coccato, L., Freudling, W., Neeser, M., & Marchetti, E. 2015, in ADASS XXIV, edited by A. R. Taylor, & E. Rosolowsky, vol. 495 of ASP Conf. Ser., 4. eprint: arXiv:1411.0996

Banse, K., Ballester, P., Izzo, C., Jung, Y., Lundin, L. K., McKay, D. J., Modigliani, A., Palsa, R. M., & Kiesgen, M. 2004, in ADASS XIII, edited by F. Ochsenbein, M. G. Allen, & D. Egret, vol. 314 of ASP Conf. Ser., 4. eprint: arXiv:1411.0996

Freudling, W., Romaniello, M., Bramich, D. M., Ballester, P., Forchi, V., Garcia-Dablo, C. E., Moehler, S., & Neeser, M. J. 2013, Astronomy and Astrophysics. eprint: arXiv:1311.5411

Modigliani, A., Mulas, G., Porceddu, I., Damiani, F., & Banse, B. K. 2004, The Messenger, 118, 8. eprint: 2004Msngr.118....8M

Sacco, G. G., et al. 2014, Astronomy and Astrophysics. eprint: arXiv:1403.4865

Vernet, J., et al. 2011, Astronomy and Astrophysics. eprint: arXiv:1110.1944

Astronomical Data Analysis Software and Systems XXXV
*ASP Conference Series, Vol. **VOL** Compiled: Tuesday 30th June, 2026@ 01:30*
K. Polsterer, D. Bohlender, S. Gwyn, A. Hincks and P. Teuben, eds.
*© **YEAR** Astronomical Society of the Pacific*

ZZZDEBUG!215.tex
Verbena, J. L.|textbf
Buchbender, C.

A Robust Snakemake Pipeline for Reducing Complex CII Datasets: The M51 Case Study

J. L. Verbena, and C. Buchbender

1. Physikalisches Institut, Universität zu Köln, NRW, Germany;
verbena@ph1.uni-koeln.de

Abstract. In this work, we present a modular and scalable data reduction pipeline designed to process the challenging M51 [CII] dataset, observed with GREAT onboard SOFIA. The raw data comprise 125 GB and over one million spectra, collected across multiple flights and observing cycles (2016–2018). A key complication is contamination by a telluric ozone line, whose frequency shifts throughout the campaign, significantly complicating baseline characterization.

Our solution integrates CLASS, Python, and Snakemake to deliver a reproducible and portable reduction workflow. Principal component analysis (PCA) forms the backbone of our baseline correction strategy, enabling robust separation of instrumental and atmospheric features from the astronomical signal. The workflow is orchestrated by Snakemake and automated through Python wrappers, which generate tailored CLASS scripts for each flight, backend, and scan—ensuring optimal characterization of baseline components under varying conditions.

This pipeline achieves consistent, reliable data products while maximizing transparency and reproducibility, offering a powerful template for the reduction of similarly complex spectroscopic datasets.

1. Introduction

The German Receiver for Astronomy at Terahertz Frequencies (GREAT) was an advanced heterodyne receiver onboard the Stratospheric Observatory For Infrared Astronomy (SOFIA), operational from 2011 to 2022. Its observations of M51 covered two distinct far-infrared lines: [CII] and [OI].

The collected raw dataset totals more than 125 GB and includes over one million spectra across numerous flights and observing cycles (2016–2018). A major challenge arises from contamination by a telluric ozone line, whose frequency shifts throughout the observing campaigns, greatly complicating baseline identification and correction.

2. Pipeline Overview

Our pipeline is engineered with Snakemake to manage the complex reduction steps required. As analysis relies on the Grenoble Image and Line Data Analysis Software (GILDAS), the pipeline deploys Python-based interfaces to automate the creation of custom GILDAS scripts for each reduction phase.

To robustly address time-variable atmospheric ozone features, data is segmented by flight, and the corresponding regions are masked. PCA-based baseline correction

is performed for each flight and receiver, operating scan-wise to maximize parallel processing. This segmentation enables efficient computation across many smaller files, exploiting Snakemake's workflow management.

For PCA, dedicated calibration steps obtain spectra of pure sky (free of astronomical signal), which are subtracted from each other to isolate and characterize purely instrumental baseline effects.

3. Pipeline Components

- **Spectra grouping**: Multiple grouping stages are implemented, particularly for baseline PCA characterization. The reduction grouping is even more granular.
- **Quality metrics as associated arrays**: Calibration steps generate ancillary spectra, which are linked as associated arrays for rigorous quality assessment. System temperature and opacity are mapped to apply per-channel weighting during reduction.
- **Spectral statistics**: Statistical metrics, such as RMS and integrated intensity, are calculated for each spectrum. These are tabulated for informed decisions during data reduction.
- **Automatic line detection**: Source emission lines are detected automatically on spectra where this is possible for better baseline characterization. The level of automatic detection proves to perform better than visual inspection.
- **Preprocessing for PCA**: Both sky and source spectra are preprocessed for decomposition (component extraction) and subsequent correction. Outliers and spectra flagged for poor quality are excluded prior to PCA operations.
- **PCA baseline reduction**: PCA routines extract principal components for each flight and backend, isolating baseline signatures. These components are then subtracted from individual spectra for correction.
- **Weighted data mapping**: The final stage merges all processed spectra into a data cube. Weighting applies not only per spectrum (σ-weighting) but also channel-wise, based on atmospheric opacity to minimize the impact of ozone absorption.

Custom software modules supplement GILDAS for specialized operations that extend beyond the standard installation.

In Figure 1 we show an example of PCA correction acting on a set of spectra. The components are derived per flight/backend combination. In the figure, on the left side, the different components and the mean spectrum before and after correction. The differene between corrected and original spectra are shown in the middle top panel. Below the spectra before and after correction are shown in waterfall plots.

4. Current Status and Challenges

The M51 [CII] data reduction is ongoing. Main challenges include dataset size and the many parameter choices for PCA-based baseline correction, resulting in a non-unique solution space that is complex to explore and optimize.

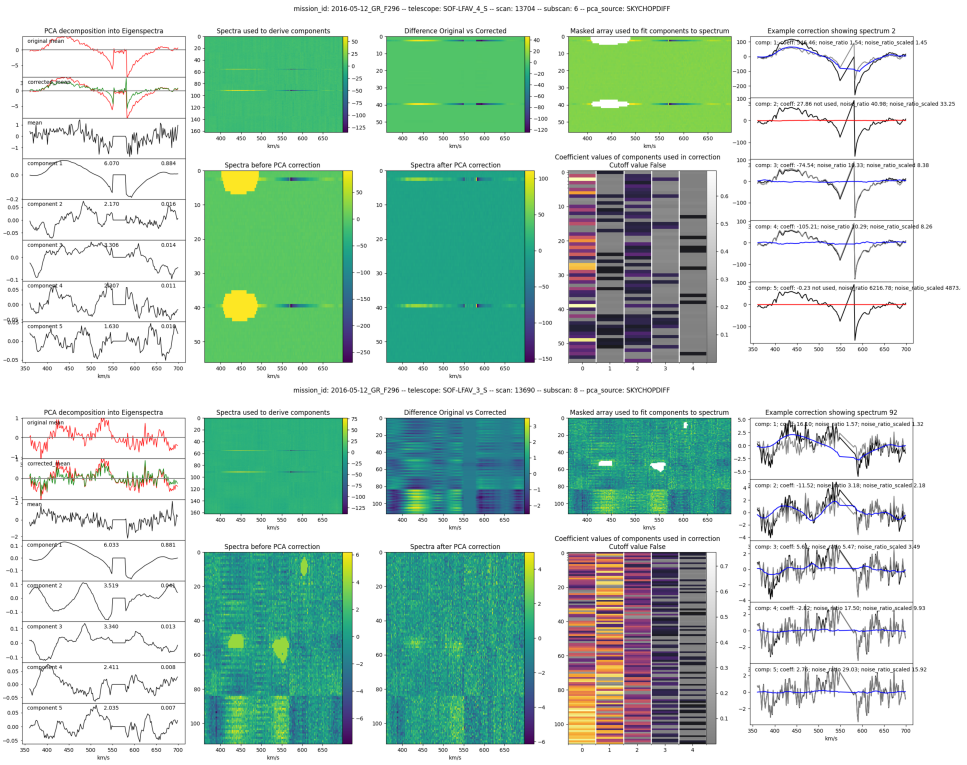


Figure 1. Examples that highlight the efficacy of PCA in subtracting baseline contributions from small sets of spectra.

5. Acknowledgements

This reduction effort is part of the SOFIA/GREAT support initiative, funded under DLR grant DLR 50OK 2401.

6. References

www.upgreat.uni-koeln.de

Astronomical Data Analysis Software and Systems XXXV
*ASP Conference Series, Vol. **VOL** Compiled: Tuesday 30th June, 2026@ 01:30*
K. Polsterer, D. Bohlender, S. Gwyn, A. Hincks and P. Teuben, eds.
*© **YEAR** Astronomical Society of the Pacific*

AIPS
AIPS++
AIPS convention
CASA
CFITSIO
CHIPASS
distortion
Doxygen
FITS
FITSWCS
HEALPix
WCSLIB
CASA! ascl:1107.013
CFITSIO! ascl:1010.001
WCSLIB! ascl:1108.003
ZZZDEBUG!216.tex
Calabretta, M. R.|textbf

WCSLIB - its origins and development - ADASS XXXV software prize, invited talk

Mark R. Calabretta

Australia Telescope National Facility / CSIRO, Sydney, New South Wales, Australia; mark@calabretta.id.au

Abstract. 2025 marks thirty years since the first release of WCSLIB, a software library closely linked to FITSWCS, the FITS World Coordinate System standard. From the start, WCSLIB informed the development of FITSWCS and now provides one of several practical implementations. In this talk I will describe WCSLIB's origins, its close connection with FITSWCS, and major milestones in their evolution. I conclude with some general observations on software development as I have experienced it in relation to WCSLIB over the past decades.

1. Origins

WCSLIB is an implementation of the FITS World Coordinate System standard (FITSWCS). Its origins and evolution are deeply entwined with FITSWCS, not just in implementing the standard but also in informing and validating its development. As such, it fulfils the role of a reference implementation. Yet it is and has always been intended primarily as a software library for use in practical astronomical applications.

The FITSWCS standard as we know it today was born from AIPS++, now known as CASA, a software package intended as the successor to AIPS, the venerable Astronomical Image Processing System. The AIPS++ project was undertaken by an international consortium of radio astronomy institutions, including ATNF. Thus, I found myself as part of a team of programmers gathered at NRAO's headquarters in Charlottesville, Virginia, for the first six months of 1992 to begin development on the package. Stemming from a long-held interest in map projections, I volunteered to develop the image coordinate classes and was duly assigned the job.

At the time, the widely-used "AIPS convention", defined by the oft-cited AIPS memos 27 and 46, written by Eric Greisen (1983; 1986), specified how celestial coordinates were to be represented in FITS. It comprised eight map projections, of which four were in common use, with a formalism that did not handle oblique coordinate grids. I envisaged a wider range of projections for AIPS++ with general handling of obliquity, as elucidated in AIPS++ memo 107 (Calabretta 1992). However, I realised that such an extension would be futile if not accomodated by FITS.

By a quirk of fate, Eric had spent two years on sabbatical at ATNF in 1990 and 1991. Based at NRAO, Charlottesville, Eric was the AIPS project manager and, as second author on the paper that introduced FITS – Wells, Greisen, & Harten (1981) – was also a co-founder of FITS, the Flexible Image Transport System, adopted formally by the IAU in 1982 for general use in astronomy. As I maintained AIPS at ATNF, and we both had an interest in birding, I got to know Eric quite well while he was in Syd-

ney. In Charlottesville the following year our birding activity continued nearly every weekend with the Monticello Bird Club so that at the end of six months I had quite a decent US bird list. In fact, it exceeded my Australian list at the time. During this period Eric and I had occasionally discussed FITSWCS in general terms, and a vague notion developed that we should put our heads together and "do it properly". While the ornithological interest may seem irrelevant, in retrospect I think this secondary connection probably helped to unite us over the ten years of hard slog that it ultimately took to bring FITSWCS to fruition.

By November 1992, well after I had returned to Sydney, and following ADASS II in Boston where a special session was held on it, interest in FITSWCS had again come to the fore. With some nudging from Brian Glendenning and Peter Teuben, Eric and I embarked on a project that I naively imagined would take a year or two! At least, I thought, the timeframe would be consistent with my intentions for AIPS++.

I hurriedly completed AIPS++ memo 107 in mid-November 1992 and, collaborating with Eric by copious emails, we soon added FITS to it. This included the CD matrix formalism developed by Bob Hanisch and Don Wells (1988), that being the limit of progress on FITSWCS when we started. We announced the first draft of *Representations of Celestial Coordinates in FITS*, now known as WCS Paper II (Calabretta & Greisen 2002), for public comment on sci.astro.fits on 4 February 1993.

Already by April 1993, Rick Balsano, a summer student at LANL working under Wayne Landsman, became the first to code our FITSWCS proposal. His programming in IDL uncovered a handful of errors in the projection equations, mostly in the nature of typos, but with a couple of significant fixes. We were surprised at how quickly things were moving. Development proceeded subsequent to Eric presenting a poster paper in June at the American Astronomical Society meeting in Berkeley, with some useful feedback from the community resulting in a further three public drafts. Thereafter, Eric presented a poster and a talk at the FITS BoF at ADASS IV in Baltimore in September 1994 (Greisen & Calabretta 1995). As AIPS++ could not use the IDL implementation, it seemed about time to start work on WCSLIB.

2. Development

Early in November 1994 I started coding our FITSWCS proposal independently in Fortran 77 and K&R C. I was targetting the C library at AIPS++, which was written in C++, but with an eye to anything else that could bind to C. This was against some grumbles from within AIPS++ that it should be written directly in C++, which, I countered, would have restricted its applicability. The Fortran library was aimed at AIPS, Miriad, IRAF, GIPSY, and any others of the major astronomical data reduction packages of the time.

Being strongly influenced by its coding style, I chose the name, WCSLIB, as a nod to Pat Wallace's SLALIB, which I had studied closely in writing the ephemeris routines for the Australia Telescope Compact Array, which, likewise, I had named ATELIB.

WCSLIB 1.0 was released on 31 January 1995 under a GNU Library General Public License, in accordance with AIPS++ policy. In hindsight, the choice of LGPL, versus GPL, certainly helped to foster the adoption of WCSLIB. Later, under the influence of an article written by Richard Stallman, *Why you shouldn't use the Lesser GPL for your next library*, I switched to GPL in WCSLIB 4.0. However, this created unintended licensing problems. I am reminded particularly of a project at Brigham and

Women's Hospital in Boston that sought to adapt medical imaging software for use in astronomy. Confronted with legal issues of which I had limited interest, I was relieved to switch back to LGPL in release 4.3.

The first release of WCSLIB concerned itself solely with coding the projection equations and related spherical coordinate transformations, essentially as had been done previously in libraries that implemented the AIPS convention. In the process I uncovered several more significant corrections to the FITSWCS draft. There was little in WCSLIB 1.0 relating to FITS other than instructions on how to set the Euler angles from FITS header keywords. In fact, my intention was to make the implementation of the projection equations as general as possible so that, even today, they only acknowledge FITS's existence in setting the default value of the radius of the generation sphere to $180^\circ/\pi$, and in the three-letter projection codes. This policy carried on later in the implementation of spectral, logarithmic, and tabular coordinates, distortions, etc.

My coding of the projection equations was informed by a program I had written for my own amusement during my Ph.D. years. It plotted celestial charts with various coordinate systems and map projections, inspired by a text book I had tracked down in the geography library at Sydney University, *Map Projections*, by Kellaway (1946). In particular, I opted to use angles in degrees specifically because, for some projections, it is important to be able to represent the latitude at the pole exactly: 90° rather than $\pi/2$ radians. It also accords with the FITS prescription of measuring angles in degrees.

Reviewing it now after thirty years, WCSLIB 1.0 does seem rather rudimentary. Nevertheless, it garnered much impactful feedback. Most notably, Rick Ebert (IPAC/JPL) and Lindsay Davis (NOAO) independently reported an apparent bug that actually arose from a serious shortcoming in the FITSWCS formalism. Here Eric and I were confronted with the dreaded monster emerging from under the bed! We were in a quandry for a week or two as to what to do about it. Ultimately we decided to go the whole hog and generalise the formalism to allow the reference point to be anywhere on the sphere, in particular for the sake of the conic projections. It led to the introduction of the `LATPOLE` keyword as elucidated in Sect. 2.4 of WCS Paper II, which I consider to be the most abstruse part of FITSWCS.

WCSLIB 2.1, the next release in November 1995, through to WCSLIB 2.9, in April 2002, progressively implemented the rest of the celestial coordinates part of the FITSWCS proposal, including developments within that period, which it validated. WCSLIB 2.1 began to acknowledge the existence of FITS by parsing the CTYPEia keywords. However, it was still left to the applications programmer to obtain the WCS keyvalues from the FITS header and plug them into disparate parts of WCSLIB.

By WCSLIB 2.4, Ray Plante (NCSA) had translated the C library into Java, which was included in that release only, and Brian Glendenning (NRAO) had written the C++ class wrappers for AIPS++.

Also in this period, WCSLIB 2.5 was joined by a curvilinear coordinate plotting routine, PGSBOX, ultimately intended to become part of PGPLOT. While general in nature, it came with an interface to WCSLIB for plotting celestial coordinate graticules. It was written expressly for AIPS++ and proved to be a devilishly tricky piece of coding. However, while it was absorbed into AIPS++, it never did made it into PGPLOT.

Our FITSWCS proposal was split into two papers in 1999, with Eric taking Paper I while I took Paper II. They were accepted together for publication on 9 September 2002 and ratified as a FITS standard that December (Greisen & Calabretta 2002; Calabretta

& Greisen 2002). This provided an impetus to reconsider the design of WCSLIB at a fundamental level.

I had coded WCSLIB 1.0 first in Fortran 77 and then translated it to an independent C implementation as I figured it would be harder to do the other way around. However, Fortran severely restricted what could be done in C, which I had always considered to be the primary implementation. By WCSLIB 2.9 this had become such an impediment, not to mention irritation, that I decided to abandon the independent Fortran implementation and replace it with a set of wrappers on the C library. This would give me the freedom to design the various WCSLIB data structs as I saw fit. Considering what they have to do, the Fortran wrappers, which are written in C, continue to amaze me at how well they work.

At the same time I had been trying to persuade Richard Gooch (ATNF) to use WCSLIB to add FITSWCS support to his *Karma/kvis* package, *kvis* being a visualisation tool developed in-house by ATNF. Richard, being a committed speed daemon, said he would consider it but only if the API was vectorised. The idea appealed to me and I decided to add it to the mix of a major revamp. However, WCSLIB never did find its way into *Karma*.

Thirdly, I had decided to give WCSLIB a FITS header parser as the next logical progression and consequently the `wcsprm` struct would have to hold the keyvalues that it extracted. But that also meant that WCSLIB would have to do its own memory management.

Finally, by this time WCS Paper III (Greisen, Calabretta, Valdes, & Allen 2006) was in an advanced state, including its specification of tabular coordinates, and I had written several drafts of WCS Paper IV on distortions (Calabretta et al. 2004a,b). Consequently, I had a pretty good idea of what would be required of WCSLIB to implement them. The changes would necessarily affect the API, which I realised would be painful for WCSLIB's users. However, I decided it would be best to get it over and done with in one hit. Perl, Python, Qt, and many others have gone through similar upheavals. I prereleased the header files and test programs in December 2002 and, following positive feedback from Jessica Mink (CfA), a key user, issued WCSLIB 3.0 as a beta release on 1 April 2003, which seemed an appropriate date! WCSLIB 3.1 followed at the end of that month.

Parsing WCS keyrecords from FITS headers was the next notable advance, the difficulty here being that WCSLIB doesn't and won't ever read FITS files, so needs assistance from some other piece of software. As it happened, the IAU General Assembly was held in Sydney in July that year, 2003, and I found myself discussing it with Bill Pence. He agreed to provide a function in CFITSIO to scoop up a whole header in a character array, which an applications programmer could then pass to WCSLIB to sift out the WCS goodies. I coded the header parser, *wcspih()*, entirely in *Flex*, an entertaining exercise for a novice, and quite sufficient for FITS's simple *"keyword = keyvalue"* syntax. It appeared first in WCSLIB 3.5, released in June 2004.

In retrospect, this was a watershed in that it allowed an applications programmer to compute world coordinates for a FITS image without needing to know anything about what was in the header or, indeed, anything much about FITSWCS. Further parsers followed in due course: for parsing and converting units, and a completely generic header parser, *fitshdr()*, not related to FITSWCS, which I was asked to write for the benefit of AIPS++. A parser for binary table headers, including pixel lists, *wcsbth()*, a

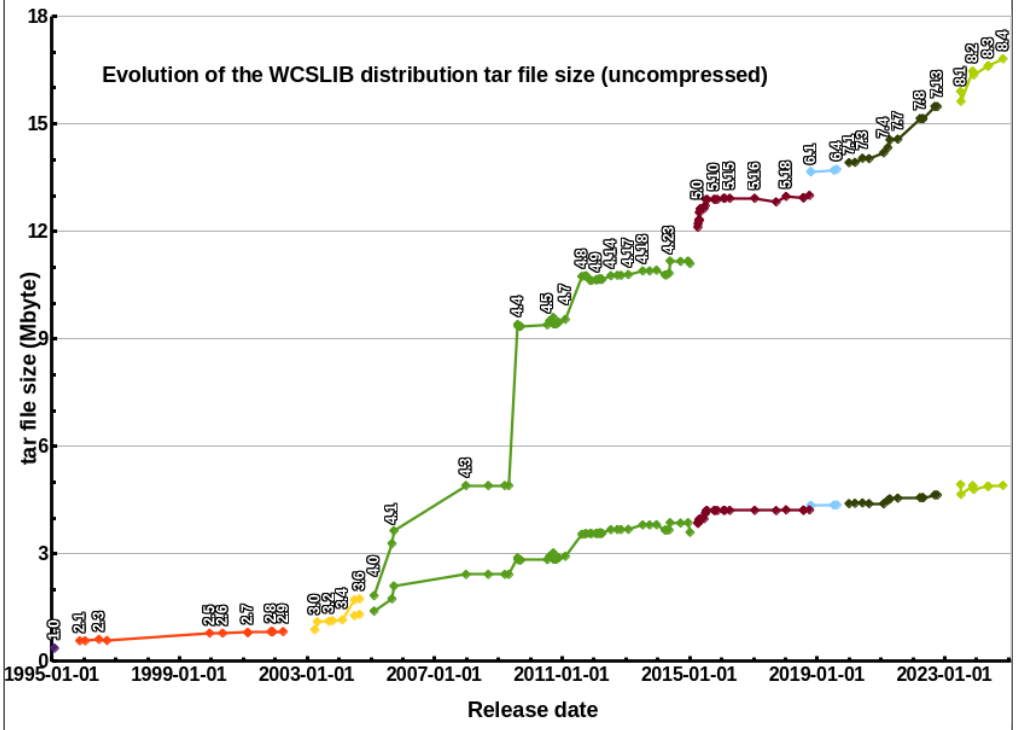


Figure 1. Evolution of the WCSLIB distribution tar file size (uncompressed). Release 2.4 excludes a translation of the C library into java by Ray Plante (NCSA). The large jump at WCSLIB 4.4 was due to the inclusion of the *Doxygen*-generated user manual in HTML and PDF formats, whereas 4.1 and 4.3 were boosted by preprocessed *Flex* code. The lower trace omits these extras, thereby offering a better estimate of the true size of the code base. Other jumps are identified in the text. Omitting the *Doxygen* and *Flex* output, WCSLIB 8.4 is ×13 larger than 1.0. It came as quite a surprise to me when I prepared this graph to see the rate at which WCSLIB is still steadily accreting code.

fair bit more complicated than image headers, eventually appeared in WCSLIB 4.3 of December 2007.

The first few releases of WCSLIB 4.x, dating from 2005, saw a spurt of progress, in particular in implementing tabular coordinates. Milestones were:

- 4.0 (2005/02/07): HEALPix projection, `HPX`, implemented, validating WCS Paper V, *Mapping on the HEALPix Grid* (Calabretta & Roukema 2007).
- 4.1 (2005/08/31): Implementated tabular (`-TAB`) coordinates (Calabretta 2006).
  Conversion from K&R C to ANSI C89 completed.
- 4.2 (2005/09/23): Installation process brought under GNU *autoconf*.
- 4.3 (2007/12/27): *wcsbth()*, binary table image array and pixel list header parser.
  wcshdo() writes out a `wcsprm` struct as a FITS header.
  Utility programs *wcsgrid*, *HPXcvt*, and *fitshdr*.

- 4.4 (2009/08/10): Generation of WCSLIB user manual from header prologues using *Doxygen*.

  Utility program *wcsware*.

- 4.8 (2011/08/15): Error diagnostics mechanism, contributed by Michael Droettboom (STScI).

- 4.18 (2013/07/13): Butterfly projection, `XPH`, implemented, validating WCS Paper VI (Calabretta & Lowe 2013).

Regarding tabular coordinates, again WCSLIB relies on external software to read the necessary information from a FITS binary table extension. However, this was rather more complicated than simply reading a header into a character array and necessitated providing a function, *fits_read_wcstab()*, for inclusion in CFITSIO or to act as a programming template for other FITS readers. Another challenge here was to keep WCSLIB and CFITSIO independent of one another, which in practice meant not including each other's header files. I described the work in a poster paper at ADASS XV in San Lorenzo de El Escorial (Calabretta 2006).

WCSLIB was, and still is documented primarily by plain-text prologue comments in its header files. These are easy to maintain as a natural part of code development – good, up-to-date documentation being a crucial part of any software package. In adapting document generation to *Doxygen*, which undoubtedly improved things for applications programmers, I didn't want to clutter the code with document markup. However, now with experience in writing the FITS header parsers, I realized that it could be done essentially without any markup by strictly formatting these prologue comments. A *Flex* parser would then be able to extract them and insert the appropriate *Doxygen* markup as part of the document compilation process. The parser, *doxextr*, is augmented by simple sed scripts that control the font used for particular keywords or phrases. Looking now at the output from *doxextr*, blighted as it is by document markup, there is no way that I would want to work directly with it.

After retiring in 2012, while continuing to maintain WCSLIB, I mostly spent the first 18 months completing the CHIPASS 1.4 GHz radio continuum survey, a complex data reduction project I had started years before (Calabretta et al. 2014). Once published, and also with WCS Paper VI out of the way, I was able to devote time to adding missing capabilities to WCSLIB, chief among which was the implementation of distortions, as per WCS Paper IV, which remains unpublished. This went pretty much according to plan, noting, in particular, the ease with which record-valued keywords could be parsed. However, supporting the legacy distortion functions – DSS, TPV, SIP, TNX, and ZPX, which between them cover the gamut of possible ways of defining a distortion function! – required creating a superset of all of them. Thus was born the Template Polynomial Distortion function or TPD. Distortion handling first appeared in WCSLIB 5.0 of 5 April 2015, a beta release, with refinements and developments following in eleven releases over the next six months with the help of much feedback. The formalism proposed in WCS Paper IV was thus validated as handling everything expected of it.

WCS Paper VII (Rots et al. 2015) on time-related keywords was implemented in WCSLIB 6.1, released in October 2018. It required the header parsers to recognise the new keywords, and the `wcsprm` struct to be expanded to accomodate them. Subsequently, I was asked twice separately to accomodate various non-standard keywords

used in solar astronomy, leading to WCSLIB 7.x and 8.x. Figure 1 charts the evolution of WCSLIB via the size of the distribution tar file.

3. In retrospect

In retrospect, one thing that impresses me regarding development of WCSLIB is the critical importance of user feedback. This has ranged from bug reports, sometimes with patch; requests for new functionality, often leading to new service routines; or even contributions of major sections of code, such as the error diagnostics reporting. Warnings issued from compilers or runtime analysers such as *valgrind* have been a major contributor, often reported by various Linux package maintainers, who run the latest version of the compiler with strict checking enabled on platforms that I cannot test directly. While often just nuisance warnings, occasionally they have uncovered real problems and must always be investigated. Feedback relating to thread-safety has been another major contributor.

Another important aspect of user feedback is that is has often provided me with the only indication that anyone actually uses WCSLIB! For example, when WCSLIB first implemented distortions I immediately received copious feedback, which was invaluable in honing the code, but also revealed that it was useful to many people. On the other hand, I've had little feedback over the years relating to tabular coordinates (`-TAB`) and am left wondering whether this complex piece of code gets much use. Likewise, I was not keen to add *wcshdo()* to the library as it seemed superfluous, after all if a `wcsprm` struct was constructed from a FITS header, why would you need to recreate it? However, the amount of feedback I have received tells me that it is widely used.

Feedback has also alerted me to WCSLIB wrappers in other languages, such as Python, Java, Javascript, Julia, and R, and its incorporation into other packages, such as Astropy and JuliaAstro. Likewise, its presence as a package in various Linux distros, ranging from Homebrew to Ubuntu.

In light of the pivotal role of user feedback, I felt it important to acknowledge contributors and have done so via the THANKS file included with WCSLIB, which has now accumulated 115 names. Some have made many contributions, others one or two. Contributors are also acknowledged for specific contributions via the CHANGES file, where some names will be found to occur frequently!

Of course, ideally code that is released should not contain bugs, but we all know that is impossible. From the first, WCSLIB has been distributed with a suite of test programs, which are also meant to serve as example code. They have proved essential in picking up silly, and not so silly coding errors before they get out the door. However, more subtle errors may take years or even decades to reveal themselves. For example, accounting for the many special cases in the spherical coordinate transformation equations of WCS Paper II proved to be exasperating and, nine years after first coding them, finally led me to write the mathematical notes supplement to the paper. As another example, recently it was pointed out, in essence, that a coordinate vector containing a mix of NaN and non-NaN elements should survive unchanged if multiplied by the unit matrix. However, noting that $0 \times \mathrm{NaN} = \mathrm{NaN}$, this wasn't happening, and a weird looking test was needed to account for the possibility. This was in a basic piece of code dating back to WCSLIB 2.1! While the rate of bug reports may be slowing, it seems they will never stop.

One of the biggest frustrations when writing a software library is finding that you need to change the API. There are remedies for this in C++ but in C and Fortran the only recourse is to write a new function with a different name and deprecate the old one. To do otherwise risks losing your user base. I bit the bullet at WCSLIB 3.0 but would never countenance doing so again. If I was to start again from scratch, I would keep much the same structure but alter the API in some respects, particularly relating to the error reporting mechanism and in extracting most of the auxiliary data from the `wcsprm` struct.

Another thing that strikes me about WCSLIB is having to support non-standard usage. The history of FITSWCS, particularly its long gestation period, allowed what is now errant usage to gain a foothold and there is an expectation that WCSLIB will accomodate it. My approach has been to quarantine non-standard usage as much as possible by providing WCSLIB with a suite of routines that translate non-standard constructs at as early a stage as possible so that the rest of the library doesn't need to deal with them.

Technological change over the past 30 years has been relentless and has affected WCSLIB in many ways. Compilers have become cleverer and cleverer, mostly for the better, but sometimes I think too clever for their own good! While I embraced selected features of C99, Link Time Optimisation (LTO) essentially forced Fortran 2003 onto the Fortran 77 wrappers at WCSLIB 8.3 in specific cases, as required by some Linux distros. This greatly complicated the wrappers. As compilers have become better at finding potential problems with code I feel they have also become more prone to producing false alarms. Annoyingly, these "cry wolf" warnings must be fixed lest they obscure signficant warnings; the goal has always been that WCSLIB should compile cleanly, that is, with strict warnings enabled but without producing any.

You may wonder what motivated my interest in WCSLIB. Initially, as I related, FITSWCS and WCSLIB were a part of AIPS++ that I had responsibility for. When the Virtual Observatory concept was taking off in the 1990s, it was understood that such standards and the software to implement them would be of key importance and ATNF was keen to be involved. When I left AIPS++ in 1998, I took over the software used for reducing data from the 21 cm Parkes Multibeam system, of which I was the sole maintainer until retirement. Circa 2008, I became interested in mining the HIPASS spectral line data set to construct a continuum map of the southern sky. As part of the process I added full WCS capability to our map-producing gridder named *gridzilla*. The ability to grid onto a few well-chosen map projections proved to be a key part of the data analysis, ultimately resulting in CHIPASS. Later, I was given responsibility for maintaining ATNF's fork of Miriad, which, I decided, would benefit from full WCS capability. Inevitably, this motivated a refinement of WCSLIB's Fortran wrappers. Truely, there is no substitute for using your own code to discover, and repair, its shortcomings.

I've maintained WCSLIB since retiring in 2012 on a volunteer basis, essentially as a hobby and a way of keeping in touch with current activity in astronomy. Over the years, through FITSWCS and WCSLIB, I've corresponded with a wide gamut of astronomers, from grad students to old-timers, including a Nobel Prize winner. I've occasionally thought about divesting WCSLIB and looked briefly at putting it on Github. However, over time the code base has accumulated a wealth of RCS check-in comments, which I am meticulous about and often consult, and I couldn't immediately see a way to retain them. Also, I develop on RCS branches and have devised an automated release procedure, such that the RCS revision matches the WCSLIB release number. I

couldn't see a way to maintain that or anything close to it. Consequently, I put it aside, at least for now.

My involvement with FITSWCS and WCSLIB has been challenging at times, but always stimulating and certainly rewarding in many ways. Between them, they've consumed a decent chunk of my professional career, and quite a bit of my retirement as well! I'd like to thank the ADASS community for nominating WCSLIB, and the ADASS Program Organising Committee for selecting it for this year's software prize. The award came very much out of the blue and I have to say I am honoured to be included in the illustrious list of past recipients: CFITSIO, IRAF, Astropy, ADS, and SAOImageDS9.

References

Calabretta, M. R. 1992, AIPS++ Memo No. 107: Spherical Map Projections. URL http://aips2.nrao.edu/docs/memos/107.ps

— 2006, in Astronomical Data Analysis Software and Systems XV, edited by C. Gabriel, C. Arviset, D. Ponz, & S. Enrique, vol. 351 of Astronomical Society of the Pacific Conference Series, 591

Calabretta, M. R., & Greisen, E. W. 2002, A&A, 395, 1077. eprint: astro-ph/0207413

Calabretta, M. R., & Lowe, S. R. 2013, PASA, 30, e050. eprint: 1308.6647

Calabretta, M. R., & Roukema, B. F. 2007, MNRAS, 381, 865

Calabretta, M. R., Staveley-Smith, L., & Barnes, D. G. 2014, PASA, 31, e007. eprint: 1310.2414

Calabretta, M. R., Valdes, F., Greisen, E. W., & Allen, S. L. 2004a, in Astronomical Data Analysis Software and Systems (ADASS) XIII, edited by F. Ochsenbein, M. G. Allen, & D. Egret, vol. 314 of Astronomical Society of the Pacific Conference Series, 551

— 2004b, Representations of distortions in FITS world coordinate systems, draft dated 2004-04-22. URL http://www.atnf.csiro.au/computing/software/wcs/dcs_20040422.pdf

Greisen, E. W. 1983, AIPS Memo No. 27: Non-linear Coordinate Systems in AIPS. URL http://www.cv.nrao.edu/fits/wcs/aips27.ps.Z

— 1986, AIPS Memo No. 46: Additional Non-linear Coordinates. URL http://www.cv.nrao.edu/fits/wcs/aips46.ps.Z

Greisen, E. W., & Calabretta, M. R. 1995, in Astronomical Data Analysis Software and Systems IV, edited by R. A. Shaw, H. E. Payne, & J. J. E. Hayes, vol. 77 of Astronomical Society of the Pacific Conference Series, 233

— 2002, A&A, 395, 1061. eprint: astro-ph/0207407

Greisen, E. W., Calabretta, M. R., Valdes, F. G., & Allen, S. L. 2006, A&A, 446, 747. eprint: astro-ph/0507293

Hanisch, R. J., & Wells, D. C. 1988, World Coordinate Systems Representations Within the FITS Format. URL http://www.cv.nrao.edu/fits/wcs/wcs88.ps.Z

Kellaway, G. P. 1946, Map Projections (Methuen & Co. Ltd., London, England)

Rots, A. H., Bunclark, P. S., Calabretta, M. R., Allen, S. L., Manchester, R. N., & Thompson, W. T. 2015, A&A, 574, A36. eprint: 1409.7583

Wells, D. C., Greisen, E. W., & Harten, R. H. 1981, A&AS, 44, 363

Astronomical Data Analysis Software and Systems XXXV
*ASP Conference Series, Vol. **VOL** Compiled: Tuesday 30th June, 2026@ 01:30*
K. Polsterer, D. Bohlender, S. Gwyn, A. Hincks and P. Teuben, eds.
*© **YEAR** Astronomical Society of the Pacific*

ZZZDEBUG!217.tex
de
Santayana, R. P.|textbf
Lutzgendorf, N.
Palacin, H.
Basset, A.
Wasiak, F.
Littenberg, T.
FOOBAR!conference!ADASS 2019
FOOBAR!organisations!ASP
FOOBAR! ascl:1101.010

The LISA Science Ground Segment: A Distributed, Multi-Agency Infrastructure for Gravitational Wave Science

R. Pardo de Santayana,[1] N. Lutzgendorf,[2] H. Palacin,[3] A. Basset ,[3] F. Wasiak ,[4] and T. Littenberg [4]

[1]*ESA/ESAC, Madrid, Spain*

[2]*ESA/ESTEC, Noordwijk, Netherlands*

[3]*DDPC/CNES, Toulouse, France*

[4]*NASA-SGS/MSFC, Huntsville, Alabama, USA*

Abstract. The Laser Interferometer Space Antenna (LISA) is a joint European Space Agency-–NASA mission that will open a new observational window on the Universe through space-based gravitational wave astronomy (Colpi et al. 2024; Danzmann et al. 2003; Bayle et al. 2022; Baker et al. 2019). Its scientific exploitation is supported by the LISA Science Ground Segment (SGS), a globally distributed system jointly developed by ESA, NASA, and the LISA Consortium, enabling end-to-end science operations from payload commanding to data processing and long-term archiving.

Within the European SGS, the Science Operations Centre at ESAC leads science planning, near-–real-time data processing, and the generation of low-latency alerts for multi-messenger follow-up, while ensuring the curation and public dissemination of mission data through the LISA Public Archive. The Distributed Data Processing Centre (DDPC), led by CNES, coordinates a federated network of European data centres responsible for the execution of scientific pipelines and the production of higher-level data products.

Complementing these activities, the NASA Science Ground Segment (NSGS) provides independent processing pipelines, low-latency monitoring, and US-based data access and user support.

We present the overall SGS architecture, coordination model, and common infrastructure that together ensure data integrity, operational sustainability, and maximal scientific return over LISA's multi-decade mission lifetime.

1. Science Operations Centre (SOC)

The SOC acts as the central node of the SGS by intefacing between the Mission Control Center (MOC), the DDPC and NASA SGS.

On the downlink chain, the SOC supports near-real-time processing of raw telemetry into Level 1 products and the generation of low-latency alerts (within ∼1 h of an on-board trigger) to enable multi-messenger follow-up. It also manages science data archiving and distribution to SGS partners, publication through the LISA Public Archive, and the coordination of catalogue consolidation, data releases, and user support.

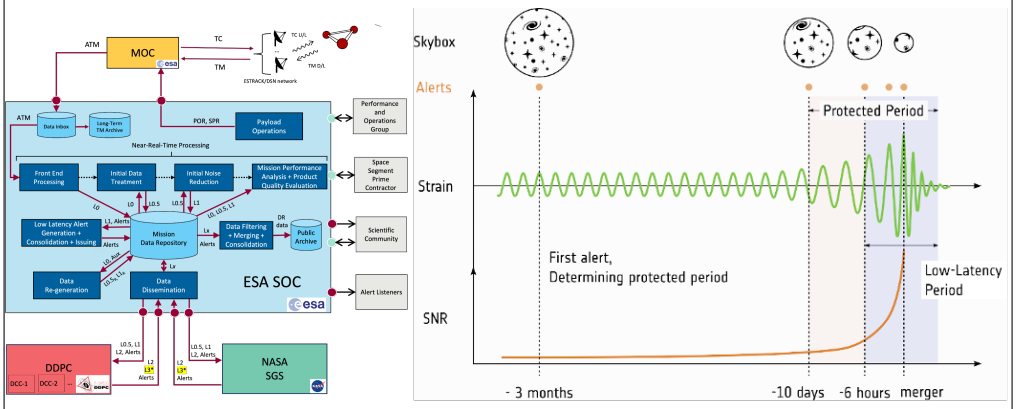


Figure 1. *Left:* SOC Architecture. *Right:* Event-driven Special Periods.

On the uplink chain, operations are designed for simplicity, with no manoeuvres or pointing and constellation orientation determined by the orbit. Routine commanding includes offset-lock planning for Doppler shift compensation and periodic payload-parameter updates. Event-driven special operational periods are enabled by near-real-time alerts generated on the downlink chain: while merger times can be predicted weeks in advance, accurate sky localisation is typically achieved only hours before coalescence. To support multi-messenger follow-up, this triggers protected periods of uninterrupted science data collection (up to 14 days) and low-latency periods with scheduled ground-station contact for near-real-time updates and rapid coordination.

2. Distributed Data Processing Centre (DDPC)

The DDPC coordinates the development and execution of European processing pipelines, including Level 0–1 processing and low-latency alert pipelines, simulation pipelines, and higher-level pipelines (e.g. Global Fit and deep-analysis pipelines) contributing to Level 2 products and Level 3 catalogue generation(Basset 2024).

The DDPC system design combines central services (global storage, metadata, and orchestration) with distributed execution across a federated network of Data Computing Centres (DCCs) contributed by multiple European partners.

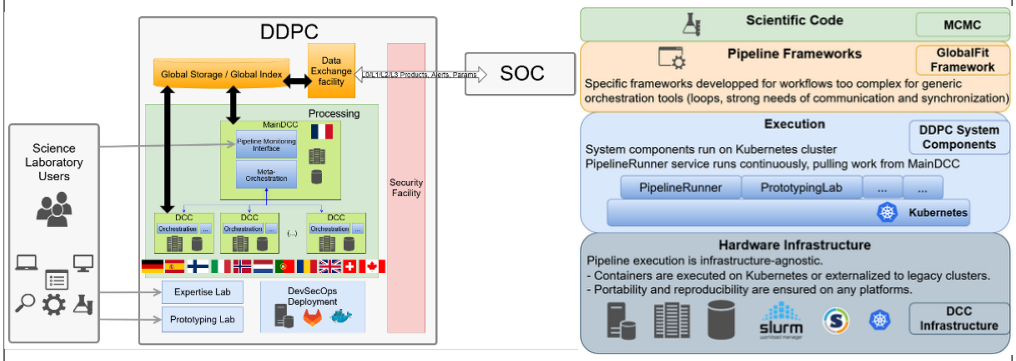


Figure 2. *Left:* DDPC architecture. *Right:* DDPC technology stack.

The DDPC meta-orchestration layer aims to optimise performance, cost, and carbon footprint through detailed resource monitoring; provide centralised monitoring and control for operators and scientists to track pipeline execution; and enable the execution of any pipeline on any compatible DCC through a cloud-native design.

3. NASA Science Ground Segment (N-SGS)

The NASA Science Ground Segment (N-SGS) provides two core functions: data processing and analysis through the NASA LISA Analysis Center (NLAC), which transforms Level 0.5 and Level 1 data into Level 2 products and delivers low-latency alerts via specialised pipelines (e.g. initial noise reduction, global fit, online analysis, and contributions to catalogue production); and science support and community engagement through the NASA LISA Science Center (NLSC), providing US-based data access, scientific user support, and a guest investigator programme.

NASA GSFC provides overall N-SGS management, while MSFC leads the development and operations of the NLAC. The NLSC is planned to be selected via a NASA Announcement of Opportunity.

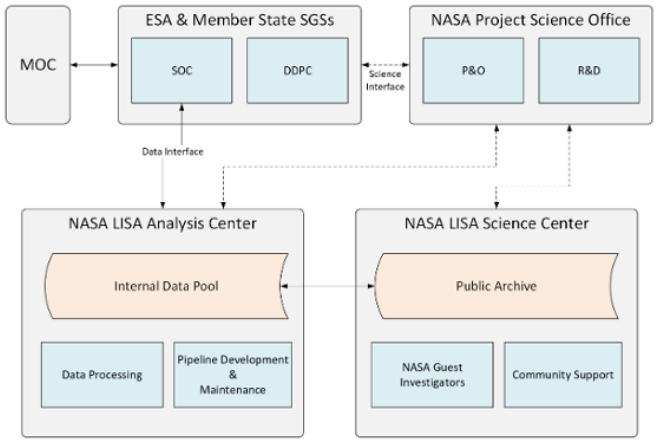


Figure 3. NSGS Architecture.

4. Common Tools and Infrastructure

ECSS-aligned quality processes and tooling are established early (PA/QA; ~250 users), covering licensing and IPR, code quality, CI/CD, V&V, and management of documentation, requirements, configuration, and risk.

To support these activities, ESA provides a set of common tools to the SGS, including GitLab for software development, version control, and CI/CD; the S2E2 Atlassian suite for collaboration, documentation, planning, action tracking, requirements, and risk management; ESA Datalabs as a community-oriented code-to-data environment; and shared ground-segment building blocks such as the Parameter Database (PDB), the Common Data Model (CDM) tools, and an associated Software Development Kit (SDK).

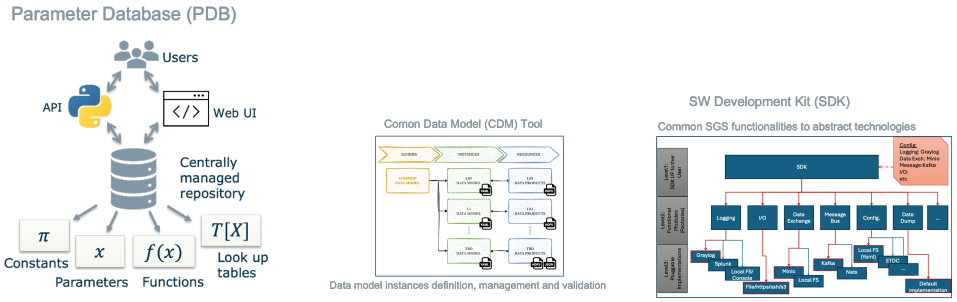


Figure 4. Common software tools.

5. Conclusions and Outlooks

LISA's Science Ground Segment (SGS) is a long-term collaboration between ESA (SOC), European member-state contributions (CNES-led DDPC), and NASA (N-SGS). While not a "big data" mission by volume, LISA requires intensive computation for source recovery (notably global-fit analyses) and sustained software evolution over multiple decades.

The SGS is built around a cloud-native approach to support flexible execution across geographically distributed DCCs, with an emphasis on scalability, modularity, performance, and maintainability. Common collaboration and quality tools are established early to support configuration control and ECSS-aligned processes. The architecture is designed to evolve with mission needs, accommodating new tools, pipeline improvements, and unforeseen discoveries throughout the mission lifetime.

References

Baker, J., Bellovary, J., Bender, P. L., et al. 2019, The laser interferometer space antenna: Unveiling the millihertz gravitational wave sky. eprint: 1907.06482, URL `https://arxiv.org/abs/1907.06482`

Basset, A. 2024, in ADASS XXXIV

Bayle, J.-B., Bonga, B., Caprini, C., Doneva, D., Muratore, M., Petiteau, A., Rossi, E., & Shao, L. 2022, Nature Astronomy, 6, 1334

Colpi, M., Danzmann, K., Hewitson, M., et al. 2024, Lisa definition study report. URL `https://arxiv.org/abs/2402.07571`

Danzmann, K., Team, L. S., et al. 2003, Advances in Space Research, 32, 1233

Astronomical Data Analysis Software and Systems XXXV
*ASP Conference Series, Vol. **VOL** Compiled: Tuesday 30th June, 2026@ 01:30*
K. Polsterer, D. Bohlender, S. Gwyn, A. Hincks and P. Teuben, eds.
*© **YEAR** Astronomical Society of the Pacific*

ZZZDEBUG!219.tex
Adrian P.
Schirra, A. S.|textbf
FOOBAR!conference!ADASS
2019
FOOBAR!organisations!ASP
FOOBAR! ascl:1101.010

Simulating antenna voltages for radio observations

Adrian P. Schirra[1,2], Anton Nöbauer[1], Savin Varghese[1], Isabel Kraus[1], Torsten Enßlin[1]

[1]*Deutsches Zentrum für Astrophysik, Postplatz 1, 02826 Görlitz, Germany*
[2]*Institute of Nuclear and Particle Physics, TUD Dresden University of Technology, 01062 Dresden, Germany;* adrian.schirra@dzastro.de

Abstract. New large radio interferometers will require the development of novel data processing algorithms. A simulator of the observed radio signals at antenna level facilitates the development and testing of some of these algorithms. We present an implementation of a simulator that produces raw antenna voltages based on a given sky model and interferometer configuration. The simulated sky model components can include point sources and pulsars. They are specified by their positions, brightness matrices, and spectra. To check for consistency, we correlate the simulated voltages and image the resulting visibilities. We compare our results with the results of the visibility simulator OSKAR for the same sky model and find a good agreement.

1. Introduction

Upcoming radio interferometers like the Square Kilometer Array (SKA) will change the way observational radio data is accessed due to the enormous data sizes involved. Research in data processing algorithms is important to ensure that the maximum amount of information is obtained from the observations under strong time and storage constraints. Such algorithms do not necessarily start after the correlation step and operate on the visibilities. Rather, they can also require the use of antenna-level data. Possible advantages of the use of antenna-level data include direct detection and mitigation of radio frequency interference (RFI) and detection of burst signals of astrophysical origin. To investigate these and similar approaches in detail, it is helpful to simulate the antenna voltages expected from a given radio sky directly. We present an antenna voltage simulator that produces the expected antenna voltages from a given radio interferometer configuration and a given sky model consisting of point sources and simplified pulsar sources (see Fig. 1). The unique feature of our simulator is the access to the raw antenna voltages which is a special requirement for the development and testing of some data processing algorithms for radio interferometers.

2. Theory

A simulated source is characterized by its sky position, Stokes parameters, and spectral parameters. The antenna signal is calculated as the inverse fast Fourier transform (FFT) of

$$F_a(\nu) = A_a(\nu) \exp(-2\pi i \nu \Delta_a / c), \quad (1)$$

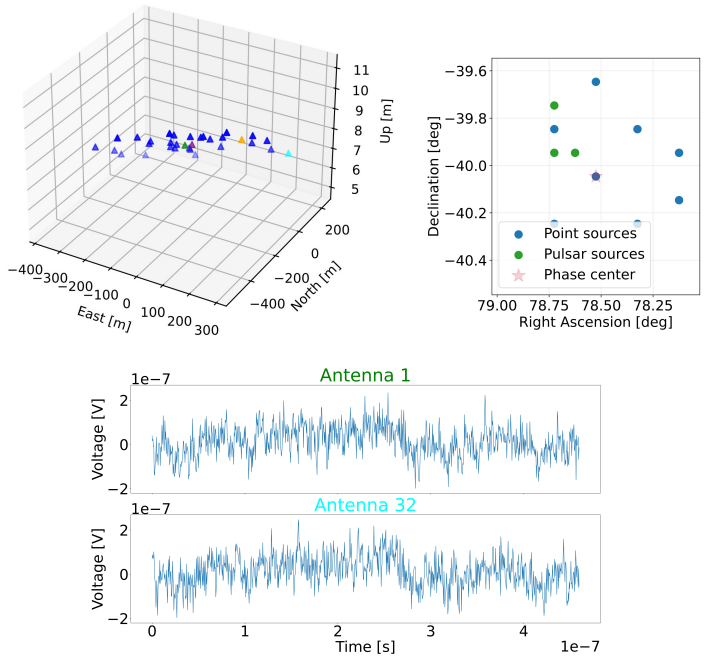


Figure 1. *Top left:* Example interferometer configuration with 32 antennas. *Top right:* Example sky model with eight point sources and three pulsar sources. *Bottom:* Resulting simulated antenna voltages.

with

$$A_a(\nu) = \left(\frac{\nu}{\nu_{\text{central}}}\right)^{\alpha/2} R(\psi_a)\sqrt{\frac{1}{2}B\xi}, \tag{2}$$

and the delay

$$\Delta_a = u_a l + v_a m + w_a(\sqrt{1 - l^2 - m^2} - 1). \tag{3}$$

Here, a is the antenna index, B is the brightness matrix, determined by the Stokes parameters of the source that describes the covariance of the sources' electrical field. $R(\psi_a)$ indicates the parallactic angle rotation (to transform the electrical field vector of each source to the local horizontal coordinate system where the polarization is measured by the antennas), ν_{central} is the frequency of the specified brightness matrix values, α is the spectral index of the source, and ξ are random complex vectors with two elements for the two polarizations.

To model simplified pulsar sources, we add the periodic suppression

$$S_a(t) = \frac{1}{2}[1 + \sin\left(\frac{2\pi(t + \Delta_a)}{T}\right) + \xi_S], \tag{4}$$

where T is the periodicity, and ξ_S are random numbers.

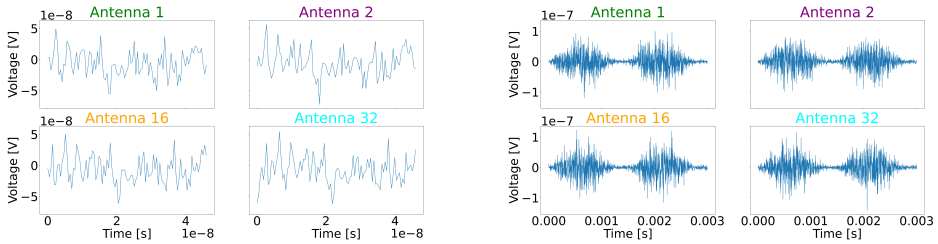


Figure 2. *Left:* Single point source indicates signal time shift due to the geometric delay. *Right:* Single pulsar source indicates periodicity.

3. Algorithm

Each simulation run is specified by a number of settings that are collected in a configuration file. They include sky model, interferometer configuration, and observation parameters. The main part of the algorithm then loops over each source defined in the sky model carrying out the following steps:

1. All antenna delays are calculated from the u, v, w position of the antennas, and the source coordinates l, m, n (Eq. 3).
2. A random source amplitude spectrum is generated (Eq. 2).
3. Inverse FFT of Eq. 1.
4. Additional periodic suppression factor when relevant (Eq. 4).
5. The resulting voltage signal is added to the total voltage signal of the respective antenna.

Finally, noise is added to the voltage signals according to a given signal-to-noise ratio (SNR).

4. Results

The main results of the simulator are the voltage timelines at the different antennas. Individual effects become clear when we consider special cases. Fig. 2 shows the time shift due to the geometric delay in the antenna voltages when a sky model with only one point source is simulated (left) and the simulated pulsar periodicity when a sky model with only one pulsar source is used (right).

We can also proceed to channelize (i.e., calculate the FFT of chunks of the real voltage time series) and correlate the simulated voltages to obtain simulated visibilities. The left panel of Fig. 3 shows an image of our simulated visibilities using the CLEAN (Högbom 1974) algorithm.

5. Discussion and further developments

There exist several programs that simulate the behavior of radio interferometers using an assumed radio sky model. The OSKAR (Dulwich et al. 2009) simulator pro-

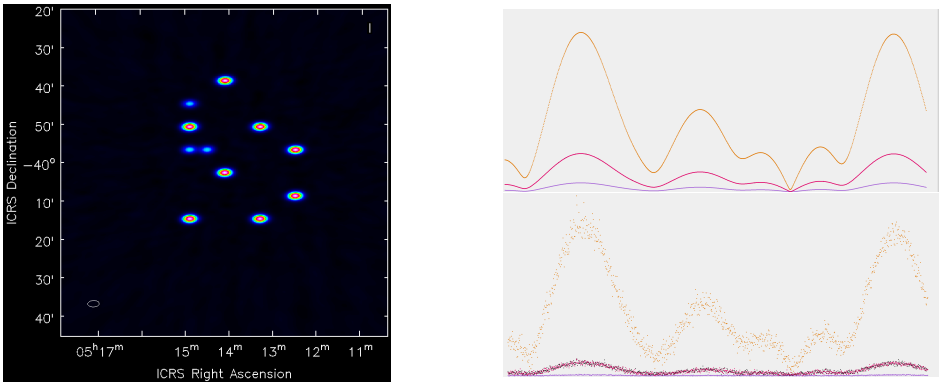


Figure 3. *Left:* Sky image using the simulated visibilities. *Top right:* OSKAR visibility amplitude as a function of frequency for one selected baseline. The colors indicate the different polarizations. *Bottom right:* Same as above but using our simulated visibilities.

duces simulated visibility data using the Radio Interferometric Measurement Equation (RIME) and offers GPU acceleration. The right panel of Fig. 3 shows the visibility amplitude as a function of frequency for OSKAR (top) and our visibilities (bottom).

Further developments include the integration of a digital twin of the receiving antennas in order to model further effects along the signal processing chain as well as the adaptation of the simulation to the planned configuration of the Lausitz Radio Array, which is a planned test array for the DSA (Hallinan et al. 2019). Other improvements concern the simulation of ionospheric effects, Faraday rotation, the mimicking of the internal delay correction applied in the F-stage of correlators, and the addition of mock RFI through specific burst time series that we want to recover to prepare related approaches for real data analysis.

References

Dulwich, F., Mort, B. J., Salvini, S., Zarb Adami, K., & Jones, M. E. 2009, in Wide Field Astronomy & Technology for the Square Kilometre Array, 31

Hallinan, G., Ravi, V., Weinreb, S., Kocz, J., Huang, Y., Woody, D. P., Lamb, J., D'Addario, L., Catha, M., Law, C., Kulkarni, S. R., Phinney, E. S., Eastwood, M. W., Bouman, K., McLaughlin, M., Ransom, S., Siemens, X., Cordes, J., Lynch, R., Kaplan, D., Brazier, A., Bhatnagar, S., Myers, S., Walter, F., & Gaensler, B. 2019, in Bulletin of the American Astronomical Society, vol. 51, 255. eprint: 1907.07648

Högbom, J. A. 1974, A&AS, 15, 417

Astronomical Data Analysis Software and Systems XXXV
*ASP Conference Series, Vol. **VOL** Compiled: Tuesday 30th June, 2026@ 01:30*
K. Polsterer, D. Bohlender, S. Gwyn, A. Hincks and P. Teuben, eds.
*© **YEAR** Astronomical Society of the Pacific*

ZZZDEBUG!229.tex
Haberhauer, F.|textbf
Buddelmeijer, H.
Leschinski, K.
observatories!ground-based
software!observation planning

ScopeSim-Targets: a unified, ESO-compatible target description framework

Fabian Haberhauer,[1] Hugo Buddelmeijer,[2] and Kieran Leschinski[1]

[1]*University of Vienna, Vienna, Vienna, Austria;*
fabian.haberhauer@univie.ac.at

[2]*Entropynaut, Groningen, Groningen, The Netherlands*

Abstract. The increasing complexity of modern astronomical instruments, and in particular ESO's Extremely Large Telescope (ELT), requires clear and reusable descriptions of observation targets across simulations, exposure time calculations, and observation planning. The absence of a standardized target description syntax currently limits interoperability between these tools and increases the risk of inefficiencies in both scientific and operational workflows. To address this, ESO has proposed a unified standard for astronomical target descriptions. In this contribution we present `ScopeSim-Targets`, a new Python package developed within the `ScopeSim` ecosystem to support this standard in simulation-driven workflows. The package provides a concise and self-consistent syntax for defining astronomical targets and scenes, enabling whole-scene simulations and the direct reuse of target definitions in ESO's proposal and preparation tools. `ScopeSim-Targets` builds on established Astropy infrastructure, supports serialization to YAML and FITS, and currently implements a range of point-source and extended target classes, with further extensions planned. The package aims to provide a common language for scientists, simulators, and observatory systems, streamlining ELT observation planning from early science cases to operational use.

1. Context and problem

In an era of increasingly complex and numerous astronomical instruments, a standardized description of observation targets becomes more and more crucial for efficient planning and archiving of astronomical observations. The ability to clearly communicate **what** is being observed with **which parameters** using simulations is crucial to avoid wasting expensive observing time that would come at the cost of other science projects. Specifically for ESO's upcoming ELT, the lack of a standardized target description syntax across observation simulations, exposure time calculations and observation planning is a major limitation.

1.1. ESO's LPO standard

To ensure seamless out-of-the-box interoperability between different systems and instruments, ESO has internally proposed a new unified standard ("Astronomical Targets at LPO") to describe astronomical target objects and scenes. The scope of this standard includes Data Flow System (DFS) tools such as the Phase 1 & 2 preparation systems, Exposure Time Calculator (ETC), Observation Preparation Tool, as well as back-end applications such as Pipelines, Quality Control and Science Archives.

1.2. Why now?

Science team members of first-light ELT instruments are currently thinking about the objects they want to observe and the GTO time that they need to allocate for them. Providing a common language to describe these objects that can be directly reused once ESO starts to plan actual observations, will streamline the whole process and avoid friction between scientists and committees. The ability to connect observation proposals to detailed simulations will minimize the failure rate of such an expensive machine as the ELT.

2. The solution in the context of the ScopeSim ecosystem

The `ScopeSim` general purpose astronomical observations simulator (Leschinski et al. 2020) aims to become an integral part of the Phase 2 preparation system in the form of an "Exposure Time Simulator" (ETS) front-end, specifically for instruments on the ELT. To support this, we are developing a new simplified target description package `ScopeSim-Targets` to become part of the overall `ScopeSim` ecosystem (Haberhauer et al. 2024). `ScopeSim-Targets` is designed to bridge the gap between `ScopeSim` and ESO's DFS infrastructure. The new package enables users to take full advantage of the enhanced whole-scene simulation capabilities provided by `ScopeSim` and subsequently directly re-use their target descriptions in ESO's proposal tools, creating a streamlined end-to-end workflow user experience. This interoperability also creates a way for both science users and operators to efficiently communicate and plan observations at such a complex machine as the ELT and its instruments.

2.1. Design considerations

The new syntax is designed to be considerably more self-consistent and straightforward from the user's perspective compared to the previously used `ScopeSim_Templates` package. On the implementation side, the new package will heavily utilize the extensive and well-tested framework provided by the `astropy.modeling` sub-package wherever applicable. Instantiation from YAML and Python should be mostly identical, serialization into both YAML and FITS headers must provide round-trip capability. From this follows a "Readme Driven Design"[1] to connect the target definition to a `ScopeSim.Source` object.

2.2. Supported target classes

The current release[2] of `ScopeSim-Targets` contains the following classes: `Star`, `StarField`, `Binary`, `Exoplanet`, `Disk`, `PlanetarySystem`, `Sersic` (simple one-component Sérsic profile galaxy). Target classes planned for future versions include: `GlobularCluster`, `YoungCluster`, IMF sampler, mock spiral galaxy, galaxy cluster, AGN host, multi-component galaxy. See also the package documentation[3] for more details and examples for each target class.

[1] see also `https://tom.preston-werner.com/2010/08/23/readme-driven-development`
[2] v0.1.2, see `https://github.com/AstarVienna/scopesim-targets/releases`
[3] `https://scopesim-targets.readthedocs.io`

3. Definition of key target attributes

All targets need a minimum of three attributes to be defined: position, brightness and spectral information. `ScopeSim-Targets` provides some default values for convenience especially in the context of simulations. In this section we provide an overview of these three attributes, including some considerations on why we choose their specific names. These attributes are sufficient to define point source targets, whereas extended targets (e.g. disks, galaxies) require some additional morphological information.

3.1. Defining target positions

Positional information is handled internally by `astropy.coordinates.SkyCoord` (Astropy Collaboration et al. 2018, 2022). Besides supplying an existing `SkyCoord` instance (useful when creating a target within a Python script), a 2-tuple of x and y offsets (in arcsec) relative to the target center, or a dictionary of more specific information is also accepted and parsed internally, see the documentation for details. The `position` and `offset` arguments are mutually exclusive. The exception to this is the `Binary` class, where `position` implicitly applies to the primary and `offset` to the secondary. Binaries with explicit positions for both components can be defined as a two-member `StarField`. If neither is specified, a (relative) position of `(0, 0)` is assumed as a default.

3.2. Defining target spectra

Any target that is not a full datacube needs some kind of spectral information. This can range in complexity from a simple stellar spectral type, for which we have basic template spectra, all the way to a user-supplied spectrum. The name *spectrum* was specifically chosen for this attribute over *SED* or any other alternatives. Any spectra should be given in rest-frame and vacuum wavelengths. All shifts are done by the tools (e.g. `ScopeSim`) using the target definitions. Line-of-sight velocity or cosmological redshift (z) can be supplied as part of the positional information, see section 3.1.

3.2.1. Possible way to specify target spectral information

- a `synphot.SourceSpectrum` (STScI Development Team 2018) instance
- a `astar_utils.SpectralType` instance (or a string able to be parsed into one), in which case the closest available template will be used.
- a `"spex:spextra_name"` string, where `"spextra_name"` can be resolved to a `spextra.Spextrum` (Haberhauer et al. 2024) database entry.
- a `"file:file_name"` string, where `"file_name"` points to a local file containing spectral information. Currently, a `"file:file_name"` identifier will be forwarded as-is to `synphot.SourceSpectrum.from_file()`, meaning any format supported by that constructor is supported here. Support for more formats is planned down the road.

3.3. Defining target brightness

All targets with separate spectral information (basically anything except a full datacube) usually also need some information on how to scale the spectrum. The only exception to this are user-supplied spectra, which are already flux-calibrated to the desired level.

3.3.1. Nomenclature

The name *brightness* was chosen as the least ambiguous of several rejected alternatives and because it matched with the ESO standard. It is not usually associated with any particular unit or physical type, but rather more of a general "how bright is my source". The following list contains rejected alternatives for naming this parameter, some of which were used in `ScopeSim_Templates`, with no particular standard:

- **Magnitude**: Physical flux units can also be used to specify brightness. Calling it *magnitude* would suggest it only accepts magnitudes.
- **Flux**: The opposite as with *magnitude*, this would suggest using a magnitude value is *not* supported.
- **Amplitude**: This name seems more fitting for the scaling of normalized profiles, can be confusing for point sources.
- **Luminosity**: Better, but still more associated with specific units, like $L_\odot$.

3.3.2. Possible ways to specify target brightness

Currently, the only implemented way to specify target brightness is the "Band + Magnitude" format, consisting of a 2-tuple with a string identifying the bandpass ("filter") in which the magnitude is given, and a numerical value or `astropy.Quantity` in `u.mag`. In this case, `mag` refers to Vega magnitudes, `ABmag` and `STmag` will be implemented in a future version. Other formats to specify target brightness which are planned for future versions include (but are not necessarily limited to): "Band + Flux", "Wavelength + Flux", and "Frequency + Flux", all of which are mentioned by ESO.

4. Installation

The package repository can be found at:
`https://github.com/AstarVienna/scopesim-targets`
The recommended way of installation is `pip install scopesim-targets`.

Acknowledgments. This work made use of Astropy a community-developed core Python package and an ecosystem of tools and resources for astronomy (Astropy Collaboration et al. 2013, 2018, 2022).

References

Astropy Collaboration, Price-Whelan, A. M., et al. 2018, AJ, 156, 123. eprint: 1801.02634
— 2022, ApJ, 935, 167. eprint: 2206.14220
Astropy Collaboration, Robitaille, T. P., et al. 2013, A&A, 558, A33. eprint: 1307.6212
Haberhauer, F., Leschinski, K., Baláž, M., Buddelmeijer, H., Czoske, O., Pott, J.-U., & Arcidiacono, C. 2024, in Software and Cyberinfrastructure for Astronomy VIII, edited by J. Ibsen, & G. Chiozzi, vol. 13101 of SPIE Conf.Ser., 1310147
Leschinski, K., Buddelmeijer, H., Czoske, O., Verdugo, M., Verdoes-Kleijn, G., & Zeilinger, W. 2020, in Software and Cyberinfrastructure for Astronomy VI, edited by J. C. Guzman, & J. Ibsen, vol. 11452 of SPIE Conf.Ser., 114521Z. eprint: 2109.13577
STScI Development Team 2018, synphot: Synthetic photometry using Astropy, Astrophysics Source Code Library, record ascl:1811.001. eprint: 1811.001

Astronomical Data Analysis Software and Systems XXXV
*ASP Conference Series, Vol. **VOL** Compiled: Tuesday 30th June, 2026@ 01:30*
K. Polsterer, D. Bohlender, S. Gwyn, A. Hincks and P. Teuben, eds.
*© **YEAR** Astronomical Society of the Pacific*

ZZZDEBUG!231.tex
Sharma, R.|textbf
Song, Y.-S.
FOOBAR!conference!ADASS 2019
FOOBAR!organisations!ASP
FOOBAR! ascl:1101.010

Reconstruction of Hubble diagram of Quasars

Ranbir Sharma,[1] and Yong-Seon Song,[1]

[1]*Korea Astronomy and Space Science Institute, Daejeon, South Korea;*
ranbirph13@gmail.com

Abstract. In this work, we create a set of algorithms to constrain the cosmological parameters using the quasar dataset. In the core of the application of quasar dataset is the non-linear relation of luminosity of X-ray and UV. In this work, we quantify the statistical behavior of the quasar parameters appears in the empirical relation of luminosity of UV and the luminosity of X-ray. We use a set of statistical techniques to validate this empirical relation, which is the basis of application of quasar dataset in Cosmology. We utilize the idea of different origins of both the quasar and the cosmological parameters to constrain the parameter space. After quantifying the strong correlation of the two quasar parameters we introduce a methodology to break this correlation and explore the model parameter space in a more efficient way. We create the Hubble diagram of the quasar dataset and constrained the cosmological parameter in the model parameter space. We test our methodology in the ΛCDM model and concretize the algorithm for the application in different dark energy models.

1. Introduction

In recent years, active work has been done to use quasars as a potential candidate for cosmic ladder. Quasars are numerous in number and can be observed at high redshifts. Also, quasars have persistent luminosities and can be observed to a redshift of $z = 7.5$. Therefore to develope a road-map of applying quasar dataset in Cosmology will be a leap in exploration of the Universe. It can be a probe intermediary to both the Cosmic Microwave Background (CMB) and Supernovae (SNIa). Quasar dataset is also a potential key to understanding the cosmological parameter tension. It can give deep insights into the actual nature of dark energy. Here, we are using the data compilation of (Lusso et al. 2020), which contains 2,420 datapoints. At the core of the application of quasar dataset is the non-linear euqation of the luminosity of UV and the luminosity of X-ray. The non-linear empirical equation is given as, $log_{10}L_X = \alpha \times log_{10}L_{UV} + \beta$. The origin of this equation (Lusso equation) is still unknown. As is shown in the recent quasar literature, there is a significant deviation from the Stanadard Model (ΛCDM) of Cosmology when we create the Hubble diagram using quasar dataset with the help of this non-linear equation. In the literature of quasar Cosmology there are three different ways to tackle this issue. First is to create a more refined quasar dataset with identification of more biases and systematics. Second is to propose a modification in the relation between luminosity of UV and luminosity of X-ray, and third is to develop a set of statistical tools for using the currently available quasar dataset to Cosmology. Here we approach the application of quasar data-set to Cosmology by developing a set of statistical tools on the basis of physical assumptions.

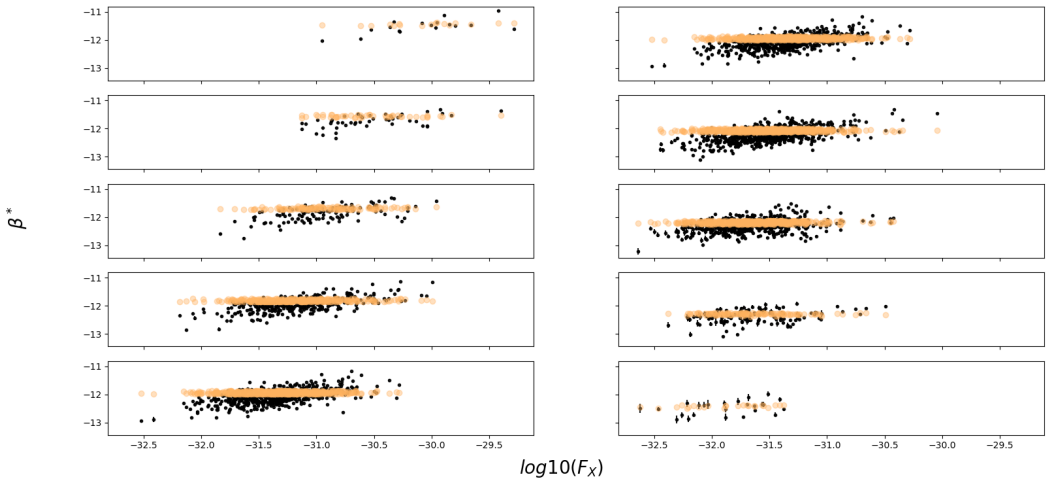


Figure 1. The black points in the figure give us the actual datapoints, whereas the orange datapoints give the theoretical prediction of β^* with $\alpha_{best} = 0.669$, and for $\Omega_m = 0.295$. This value of Ω_m comes from the χ^2 calculation for the observed dataset.

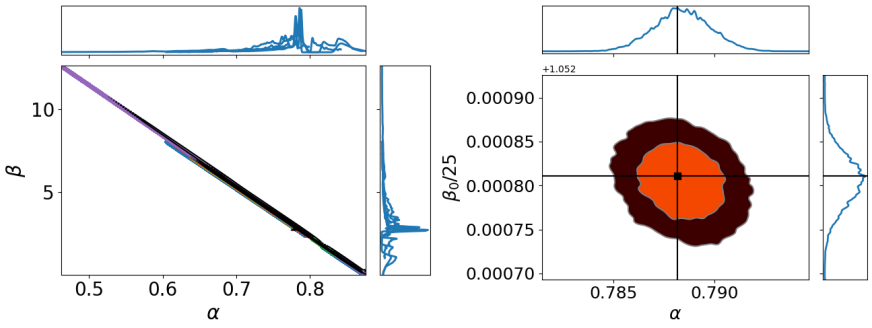


Figure 2. Here, we quantitatively showing the successful breaking of the correlation of the two parameters (α, β). In the left figure we use ΛCDM cosmology with different values of Ω_m. The fluctuation in the marginal diagram are for the MCMC chain run. Here we use the complete quasar data-set.

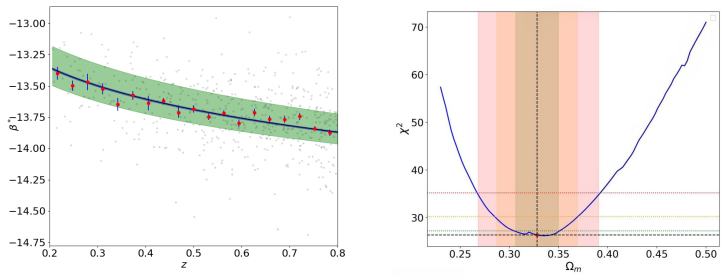


Figure 3. In this figure we use the change of parameter (α,β) parameter space and finally constrained the cosmological parameter Ω_m. In the left figure we plot the final β^* vs z curve in the observed dataset. Here we give the constrained values of the parameters. In the right figure gives the constrains in Ω_m.

This article is organized as follows, in Section 2, we describe our methodologies to refine the quasar dataset and refine the quasar data-set and consequently apply it to cosmological purpose. In section 3 we described the results and finally give our conclusion in section 4.

2. Methodology

The luminosity relation between the UV and X-ray can written in terms of flux as,

$$log_{10}F_X = \alpha \times log_{10}F_{UV} + \beta^* \tag{1}$$

In the above equation the intercept β^* is given as, $\beta^* = \beta + (\alpha - 1)log(4\pi) + 2(\alpha - 1)log(D_L)$, where, D_L is the luminosity distance. D_L is given as, $D_L(z) = \frac{c}{H_0}(1 + z)\int_0^z d_H(z')dz'$. H_0 is the Hubble constant.

We start with validifying the relation in the binned space and then we proceed to constraint the parameters. Here, we use the idea that the origin of (α, β) have different origin then the origin of the cosmological parameters. We assume a Cosmology, in our present case, it is the ΛCDM Cosmology, and we create the luminosity space of $log_{10}(L_{UV})$ and $log_{10}(L_X)$. We binned this space and we fit a straight line to find slope and intercept, which are the (α,β) of our interest.

We create the binned dataset to deal with the intrinsic dispersion. The number of bins is determined by following two criteria. Each bin should have at-least three quasar points and we neglect those bins whose intrinsic dispersion is greater than the total standard deviation of the quasar points. We use the idea that both the quasar parameters (α,β) and the cosmological parameters have different origins. We start with assuming a Cosmology and find out the luminosity space $(log_{10}(L_{UV}), log_{10}(L_X))$ from the data space of fluxes $(log_{10}(F_{UV}), log_{10}(F_X))$. We vary Ω_m and create the binned luminosity space in each case. We fit straight line in each case to get the probability density functions of (α,β) for a particular Ω_m. Therefore we have a tabulated dataset of Ω_m vs (α,β). The next step is to use the prior information of (α,β) and constrain the cosmological parameters.

3. results

With fixed Ω_m we can have a constant β^* for each redshift bin. In the fig(1), we validate the Lusso equation for the best Ω_m value. We do the same exercise for different Ω_m values. From our study it is apparent that the Lusso equation is valid for the low redshift data-points, fig(1). We also find out the strong correlation of the quasar parameters (α, β) is independent of Cosmology, fig(2). This suggest us that we can write one parameter in terms of another. We find out a way to break this strong correlation and to explore the model parameter space in a more effective way. We write β parameter in terms of α, as $\beta = \beta_0 - \alpha * \eta$, where $\eta = -30.75 \pm 1.153$. The value of η in our present case is find out only for the low-redshift quasar data points. From the fig(2) we can see that the correlation is broken successfully.

In the fig(3) we determined the Ω_m using the prior informations of (α, β). As described in the sec(2), we generate the list of dfferent values of (α, β) for different Ω_m. Giving the these values of (α, β) as is found a priori, we calculate the value of Ω_m as $\Omega_m = 0.332 \pm 0.021$.

We also complete the whole exercise in a model-independent, non-paramteric manner, using the script of (Sharma et al. 2020; Sharma & Jassal 2022). In this case we get the functional form of $\alpha(\Omega_m)$ as well as $\beta(\Omega_m)$ and give them input to the Maximum Likelihood Estimation calculation. Here we only present the model dependent, parameteric approach with the low-redshift dataset.

4. Conclusion

In the exploration to the sparse quasar dataset we find out that if the we use the idea of different origins of quasar and the cosmological parameters then along with the suitable binning technique we can apply the quasar dataset for cosmological purpose. We find out that the strong correlation of the quasar parameter in the empirical equation of luminosities is independent of cosmology. This universality of the strong correlation of (α, β) helps us to break the correlation and explore the parameter space more efficietly. Along with the parametric model-dependent approach we also use model-independent, non-parametric approach to check the validity of the Lusso equation. We apply the whole methodology to the low-redshift dataset and find out that it is consistent with ΛCDM. Now we are expanding our technique to incorporate the complete quasar dataset.

References

Lusso, E., Risaliti, G., Nardini, E., Bargiacchi, G., Benetti, M., Bisogni, S., Capozziello, S., Civano, F., Eggleston, L., Elvis, M., Fabbiano, G., Gilli, R., Marconi, A., Paolillo, M., Piedipalumbo, E., Salvestrini, F., Signorini, M., & Vignali, C. 2020, A&A, 642, A150. eprint: 2008.08586

Sharma, R., & Jassal, H. K. 2022, arXiv e-prints, arXiv:2211.13608. eprint: 2211.13608

Sharma, R., Mukherjee, A., & Jassal, H. K. 2020, arXiv e-prints, arXiv:2004.01393. eprint: 2004.01393

Astronomical Data Analysis Software and Systems XXXV
*ASP Conference Series, Vol. **VOL** Compiled: Tuesday 30th June, 2026@ 01:30*
K. Polsterer, D. Bohlender, S. Gwyn, A. Hincks and P. Teuben, eds.
*© **YEAR** Astronomical Society of the Pacific*

ZZZDEBUG!232.tex
Ivan, I. B.|textbf
Gilles, G. L.
catalogues!services!VizieR
databases!individual!Vizie
TOPCAT! ascl:1101.010
Astropy! ascl:1304.002

Re-writing the VizieR catalogue ingestion pipeline

Ivan Brossard[1], Alicia Flint[1] and Gilles Landais[1]

[1]*CDS, Observatoire Astronomique de Strasbourg, CNRS UMR 7550,France*

Abstract. Vizier provides a library of published astronomical catalogues (tables and associated data) with verified, enriched data. Since its creation in 1995, the workflow has been a semi-automated process. It makes authors' datasets compatible with the Virtual Observatory (VO) and adds additional data, such as position or links.

We rewrote the catalogue ingestion process to follow an architecture that allows for new developments while being backwards compatible with the ~30K catalogues already present in the database. Indeed, VizieR has to keep up with the evolution of the science context spurred by the dynamic VO and the Open Data era.

The new developments improve internal processes (e.g. UCD auto-detection or cross-correlation) and integrate astronomical standard such as MOCs, UAT or Datalink. They also abide better by FAIR concepts, licences, ORCIDs and linked resources.

While the original program was exclusively written in C, the new version is a mix of C and Python, written so that metadata extraction and database population are separate. The new version still uses ReadMe files for metadata and accepts multiple standards for data, such as Fits, MRT files, or csv files. These files are then formatted and ingested in PostgreSQL and in an ElasticSearch database.

In this paper, we describe the method by which we re-wrote the ingestion pipeline. Finally, we describe new ideas and features that we can now implement, thanks to the new pipeline.

1. VizieR

VizieR is a curated library of astronomical catalogues that provides enriched, verified data. Users have multiple ways of accessing the data, be it through the html pages, TAP queries (Landais et al. 2013), or interoperating with Topcat or astropy. On the internal side, Vizier works with a PostgreSQL database to store the data, as well as multiple files that describe and enrich catalogues, adding descriptions or links to other resources for instance. The internal workflow by which catalogues are ingested into VizieR is a semi-automated workflow : the description of catalogues and their enrichment with metadata are done by the documentalists at CDS, with the help of tools that streamline the process (for instance, the UCD builder, that gives proposition of UCDs for columns in the catalogue, that documentalists can verify and edit). The first part of the workflow builds the metadata around the catalogue, and relies heavily on the standardised description of the catalogue, while the second part, populating the database, needs no human intervention.

2. Original program

The original ingestion program (Ochsenbein et al. 2000) is written in C and is more than 120000 lines long. As such, maintaining the program and adding new features gets harder and harder. That is why we re-wrote the program in python as well as C, compartmentalising different parts of the workflow in different packages, which allowed us to reduce the size of the main program. This, in turn, makes the reading and understanding of the architecture easier for future developments. Indeed, the new program is in no way a simple transformation of the code from C to python.

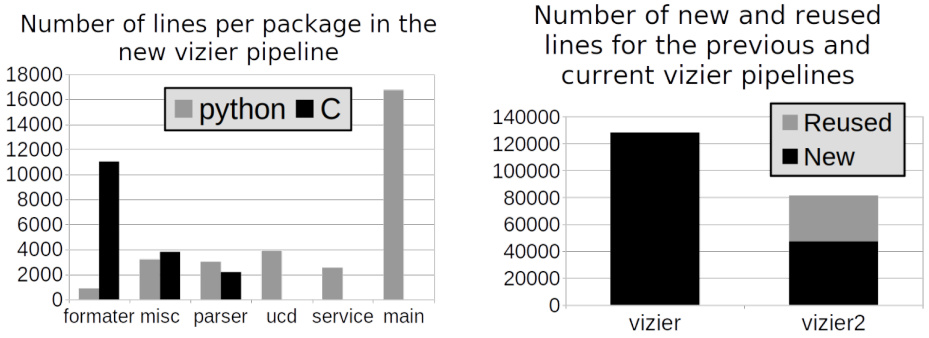


Figure 1. Number of lines per package and total for both versions of the pipeline.

3. New program

The new ingestion pipeline reuses most of the ideas behind the original pipeline. The new pipeline is subdivided in different packages for different tools. During the building of the metadata, the standardised ReadMe file is read and combined with a settings file. They are then transformed into a json file containing all metadata relevant to the catalogue. This file is then used to transform the data files themselves: adding links to other data, adding columns such as right ascension and declination, merging files, etc. Afterward, data are ingested into the PostgreSQL database (see Figure 2). A large part of the work behind writing the new VizieR pipeline has been making sure that it is backwards compatible with the ~30 thousand catalogues saved in our database. Although it mainly meant abiding by pre-existing rules, it also meant scouring the database to find exceptions, outliers, or unwritten rules.

4. New features

The catalogue ingestion program is part of a comprehensive overhaul of the VizieR service. Indeed, we added VO units, we redesigned the global search using ElasticSearch, we started using MOCS for the global indexing, and the management of large tables has been completely revised (Pineau & Landais 2023). As mentioned earlier, the new ingestion software mainly consists of a technology migration with a conservative approach. However, it includes some new features such as UAT thesauri and it will include UTF8 management and ORCID and affiliation detection. Using the new basis

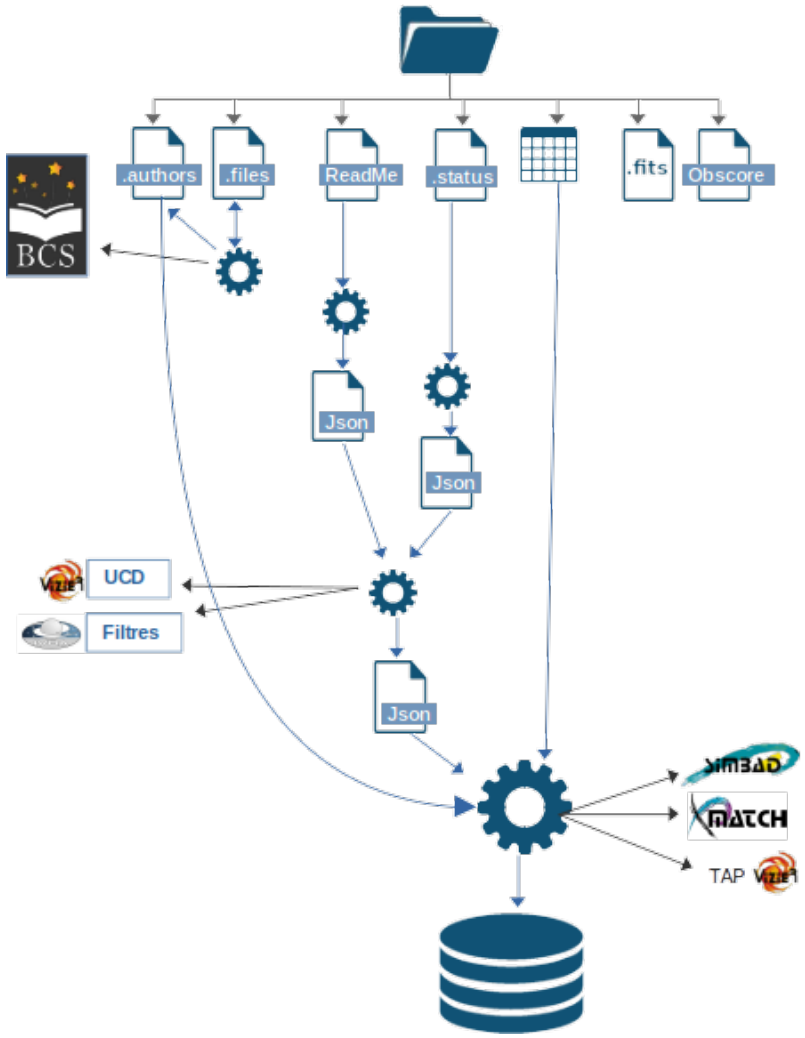


Figure 2. VizieR catalogue ingestion pipeline flowchart.

for the handling of the metadata in the VizieR pipeline, we are able to integrate new standards in the service. A long-standing issue for VizieR has been the accessibility of associated data (e.g. images, spectra) of catalogues. Indeed, apart from the visual html interface, it is impossible to access linked resources. To address this problem, we are implementing the Datalink standard (Dowler et al. 2015), by adding machine-readable links for VOTable outputs. We can add a column to the result of the user query, containing links to datalink instances, which themselves link back to the associated resources, with a short description for each.

References

Dowler, P., Bonnarel, F., Michel, L., & Demleitner, M. 2015, IVOA DataLink Version 1.0, IVOA Recommendation 17 June 2015, eprint: 1509.06152

Landais, G., Ochsenbein, F., & Simon, A. 2013, in Astronomical Data Analysis Software and Systems XXII, edited by D. N. Friedel, vol. 475 of Astronomical Society of the Pacific Conference Series, 227

Ochsenbein, F., Bauer, P., & Marcout, J. 2000, A&AS, 143, 23. eprint: astro-ph/0002122

Pineau, F., & Landais, G. 2023, in ADASS XXXIII (San Francisco: ASP), vol. TBD of ASP Conf. Ser., TBD

Astronomical Data Analysis Software and Systems XXXV
*ASP Conference Series, Vol. **VOL** Compiled: Tuesday 30th June, 2026@ 01:30*
K. Polsterer, D. Bohlender, S. Gwyn, A. Hincks and P. Teuben, eds.
*© **YEAR** Astronomical Society of the Pacific*

ZZZDEBUG!233.tex
Han, J.|textbf
FOOBAR!conference!ADASS 2019
FOOBAR!organisations!ASP
FOOBAR! ascl:1101.010

An Astronomical Dataset Management Engine for Science Platform

Jun Han[1,2]

[1]*National Astronomical Observatories, Chinese Academy of Sciences, Beijing, 100101, China;*
[2]*National Astronomical Data Center, Beijing, 100101, China*

hanjun@nao.cas.cn

Abstract. With astronomy entering an era of petabyte to exabyte scale data from next generation telescopes and surveys, existing cloud platforms face critical data management challenges. This paper present a POSIX compliant read-only file system designed for non-landing data scenarios, which can be used for the rapid construction and management of astronomical datasets. The system integrates FUSE for POSIX interface implementation, libnfs for server-side data retrieval, and a database-driven metadata layer that enables file-level access isolation through SQL-defined policies. This file system is designed to be user-centric and fully customized, delivering a "what you envision is what you obtain" experience, thus designated as YourFS. It could effectively balance accessibility and cost-efficiency for massive datasets, providing a viable paradigm for server-side data management in cloud-based science platforms.

1. Introduction

Making data open is a tradition in astronomy, and the problems of data storage and computing have been initially solved through large data center. Cyber-infrastructure is becoming a crucial part for daily scientific research in astronomy, as the datasets are too large for astronomers to download and analyze using their own computers. In terms of data management, different science platforms have proposed distinct solutions, which are technically divided into two categories. One category involves creating datasets based on scientific requirements and mounting them via links to provide data support for users' working environments (e.g. Stetzler et al. (2022); Russo et al. (2022)). The other category offers users access interfaces through an API approach(e.g. Taghizadeh-Popp et al. (2020); Nikutta et al. (2020)). Given that the vast majority of astronomical software performs data processing by directly reading files, the first method is more universally applicable. While the second method offers greater flexibility, it is not supported by most astronomical software and is thus suitable for some newly developed astronomical software applications.

For scientific users, data requirements are not static, and accessible scopes dynamically expand as data protection periods expire or varying data retrieval scopes. Considering that the vast majority of current users prefer POSIX compliance file systems and the widely adopted NFS (Network File System), this paper proposes a read-only file system with file isolation capabilities, which implements essential POSIX interfaces. It enables users to transparently mount remote NFS shares on local systems through an

interface while defining file access scopes via a database, thereby achieving seamless cross-platform data sharing. In section 2, this paper details the technical architecture and design of the file system. Section 3 introduces the software usage methods.

2. The Technical Architecture

The software is developed in C programming language, and the release version(v1.0) has been published[1]. To achieve POSIX compliance, server-side data access, and data isolation capabilities, the system employs FUSE (Filesystem in Userspace)[2] to implement POSIX interface design. It dynamically invokes libnfs[3] for data loading operations and utilizes a database to define file accessibility scopes, thereby establishing file-level access isolation mechanisms through metadata-based permission constraints. This architecture ensures both protocol compatibility and granular security control in multi-user environments. The technical architecture is showed in Figure 1.

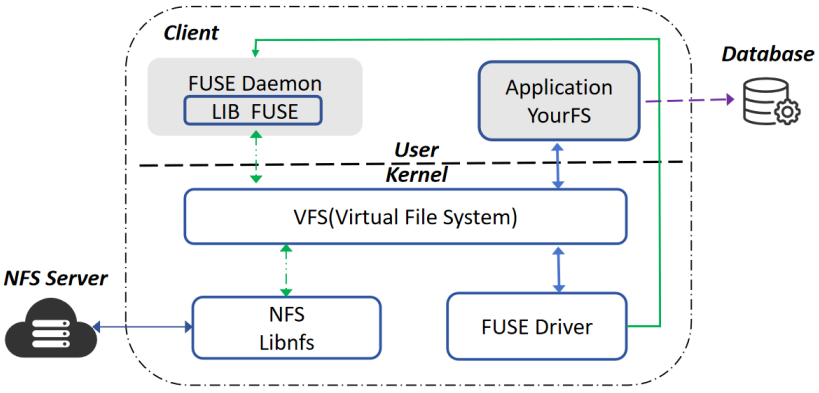


Figure 1. The technical architecture of YourFS. The proposed system ensures POSIX compliance through FUSE implementation, facilitates server-side data retrieval via libnfs integration, and implements access scope definitions through database-driven metadata constraints, achieving both protocol standardization and security enforcement.

The following describes the runtime logic when a user lists files in a directory in YourFS filesystem. When a user process(e.g., ls command) is executed, the system call readdir() to request directory contents. It is intercepted by VFS layer and forwards it to FUSE Driver. The driver encapsulates the request into a FUSE protocol message and writes it to the /dev/fuse character device. The FUSE Daemon monitors the device via libfuse, which decodes the protocol message and triggers the readdir() callback function registered in YourFS. Then YourFS executes its custom logic to retrieve directory in the database and queries metadata from a backend source via libnfs. The metadata is returned to libfuse and writed to /dev/fuse via libfuse. The driver extracts the entries,

[1] https://github.com/techwander/yourfs
[2] https://github.com/libfuse/libfuse
[3] https://github.com/sahlberg/libnfs

passes them to VFS, and VFS populates the user process's buffer. The ls command receives the entries and displays them on the terminal.

3. Usage

The implementation of this filesystem primarily involves two phases: database construction and mounting operations. To enhance clarity, three use case scenarios have been designed to demonstrate the workflow and benefits of the filesystem architecture. Figure 2 illustrates three examples of directory listing customization. The first example is used to display partial files, and it is usually used in partial data release scenario. You only need to write the file information involved to the data table. In astronomy applications, after cross-matching datasets, users often need to access only a handful of files, which are typically scattered across deeply nested directory hierarchies. This fragmented storage structure creates significant usability challenges. Example 2 demonstrates a directory restructuring example where modifying the parent directory hierarchy consolidates these files into a unified view, achieving a flat, user-friendly structure for data access. Example 3 demonstrates how users can tag special files by creating aliases. This allows files to appear under modified names or paths in the virtual filesystem without altering their physical storage locations on the NFS server. YourFS filesystem supports both SQLite3[4] and PostgreSQL[5] databases currently.

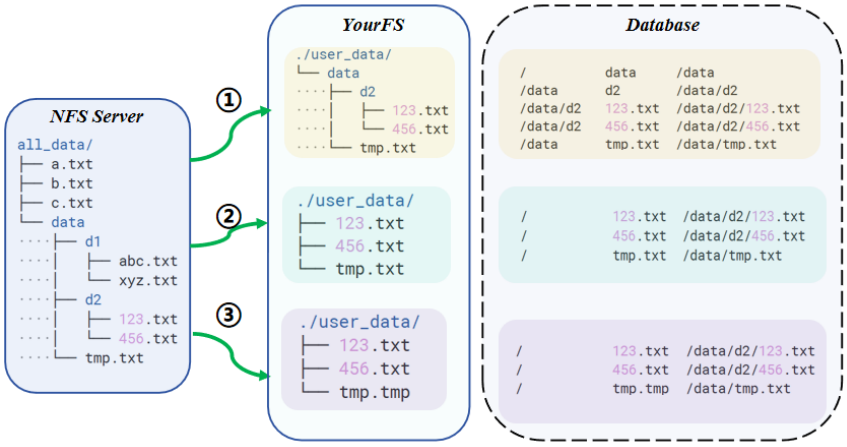


Figure 2. This figure illustrates three examples of directory listing customization. The left panel shows the file list in the NFS Server, the middle panel displays the mounted YourFS view (Partial File Display, Restructured Directory Hierarchy, Alias Creation for Specific Files), and the right panel presents the raw content stored in the database table.

[4]https://www.sqlite.org
[5]https://www.postgresql.org

4. Conclusion

The YourFS file system presents an innovative solution to the critical challenges in astronomy big data management. By integrating FUSE's POSIX compatibility with NFS's distributed access capabilities, coupled with a database-driven dynamic permission mechanism, the system provide a method for data accessibility and storage efficiency. The upcoming v2.0 release will implement advanced caching optimizations to boost software performance and significantly improve data query efficiency; it will also be made open-source and support non-relational databases such as Redis.

Acknowledgments. This work is supported by Astronomical Big Data Joint Research Center, co-founded by National Astronomical Observatories, Chinese Academy of Sciences and Alibaba Cloud. Special thanks to my son for his support. The system name "YourFS" originates from his nickname "youyou".

References

Nikutta, R., Fitzpatrick, M., Scott, A., & Weaver, B. A. 2020, Astronomy and Computing, 33, 100411

Russo, S. A., Bertocco, S., Gheller, C., & Taffoni, G. 2022, Astronomy and Computing, 41, 100648. eprint: 2209.02003

Stetzler, S., Jurić, M., Boone, K., Connolly, A., Slater, C. T., & Zečević, P. 2022, AJ, 164, 68. eprint: 2206.14392

Taghizadeh-Popp, M., Kim, J. W., Lemson, G., Medvedev, D., Raddick, M. J., Szalay, A. S., Thakar, A. R., Booker, J., Chhetri, C., Dobos, L., & et al. 2020, Astronomy and Computing, 33, 100412. eprint: 2001.08619

Astronomical Data Analysis Software and Systems XXXV
*ASP Conference Series, Vol. **VOL** Compiled: Tuesday 30th June, 2026@ 01:30*
K. Polsterer, D. Bohlender, S. Gwyn, A. Hincks and P. Teuben, eds.
*© **YEAR** Astronomical Society of the Pacific*

ZZZDEBUG!234.tex
Pineau, F. -X.|textbf
Fernique, P.
Landais, G.
Marchand, M.
FOOBAR! ascl:1101.010

On-the-fly generated HATS products

F.-X. Pineau[1], P. Fernique, G. Landais[1], and M. Marchand[1]

[1]*Université de Strasbourg, CNRS, Observatoire astronomique de Strasbourg, UMR 7550, F-67000, Strasbourg, France;*
francois-xavier.pineau@astro.unistra.fr

Abstract. We present the test developments conducted by CDS for the on-the-fly generation of catalogs in HATS format. HATS is a framework described in the recently published IVOA note "HATS: A Standard for the Hierarchical Adaptive Tiling Scheme in the Virtual Observatory." Developed as part of the LINCC project for Vera C. Rubin data, and partly inspired by the HIPS format, it basically resorts to HEALPix tessellated Parquet datasets to distribute and manipulate possibly very large tabular data.We present a prototype CDS set up to distribute VizieR large catalogues as on-the-fly generated HATS products, and discuss the advantages and drawbacks of such an approach. A similar solution can be adopted for all HEALPix indexed catalogues, benefiting from HATS aware tools without having to duplicate the data.

CDS large tables and QATSS

At CDS, each large astronomical table is stored as a single binary file sorted and indexed by HEALPix. This design allows efficient spatial access while keeping the storage model simple and easy to deploy. Representative examples of such tables include the US Naval Observatory UCAC5 catalogue, which occupies 8.9 GB, the ICCUB/AIP StarHorse 2021 catalogue at 62 GB, the ESO VHS DR5 catalogue at 708 GB, and the ESA Gaia DR3 main table, which reaches approximately 1.5 TB. Queries on these tables are performed using the *qat2s* tool, introduced at ADASS 2023. QATSS (Query Astronomical Table Serialization System) provides several key features. It is aware if HEALPix-based clustered and indexed data to enable fast positional queries. Its design is based on flat files and a single executable, which makes it easier to deploy, maintain, and mirror than solutions relying on a database management system. In addition, *qat2s* supports multi-threading and multiple input and output file formats.

The HATS format

HATS[1] provides a method for representing an astronomical table as a Parquet dataset organised according to an adaptive HEALPix grid. The dataset is laid out as a directory hierarchy, where each leaf node corresponds to a Parquet file, or tile, containing a limited number of rows. This structure makes it possible to scale to very large catalogues while remaining compatible with modern column-oriented storage and analysis tools.

[1]https://www.ivoa.net/documents/Notes/HATS/

On-the-fly HATS generation with QATSS

The objective of the on-the-fly, streamed HATS approach is to expose HATS views of CDS tables without duplicating the underlying data. Instead of precomputing and storing Parquet tiles, HATS products are generated dynamically from existing HEALPix-indexed tables. The purpose of on-the-fly, streamed, HATS products is to provide HATS views of CDS tables, without having to duplicate the data. So far, all VizieR large tables are potentially available in the HATS format. We are working on possibly adding all other tables from indexed FITS or VOTable files. At present, all VizieR tables can potentially be accessed in the HATS format using this mechanism. For regular – i.e. not "large" –tables, we rely on HEALPix sorted and indexed FITS+ files. For the current list of CDS HATS publicly available products, see: `https://vizcat.cds.unistra.fr/hats`

Implementation details

1 - HATS directory structure computation

A key constraint of the HATS specification is that each tile, represented as a Parquet file, must not exceed a given maximum number of rows. CDS large tables are already indexed by HEALPix indices ranging from order 0 to order 11, which can be interpreted as a set of density maps at increasing spatial resolutions. The HATS directory structure is computed using a recursive, top-down algorithm – see Figure 1 – applied to these HEALPix density maps. Starting from low-resolution tiles, the algorithm progressively subdivides the sky until the row-count constraint is satisfied for all tiles. For the Gaia DR3 main table, the computation of the full HATS structure typically takes around 0.06 seconds.

```
let n_srcs = get_index(order).get_cell(ipix).n;
if order == order_max || n_srcs < n_rows_max {
  res.push((order, ipix, n_srcs));
} else {
  split(order + 1, (ipix << 2) | 0, res);
  split(order + 1, (ipix << 2) | 1, res);
  split(order + 1, (ipix << 2) | 2, res);
  split(order + 1, (ipix << 2) | 3, res);
}
```

Figure 1. Recursive HATS structure generation, top-down approach

2 - Emulation of static HATS directory listings

To reproduce the behaviour of a static HATS product hosted on a conventional web server, we developed a pure Rust command-line tool – see Figure 2 – that can also be deployed as a CGI. This tool implements the various actions expected when navigating a HATS directory tree, including the generation of the root web page for a HATS product and its sub-directories, the production of the *hats.properties* file, and the generation of skymaps and related auxiliary metadata.

The correspondence between HATS URLs and CGI commands is implemented using Apache rewrite rules, examples of which are shown in Figure 3. This mechanism allows HATS-aware clients to interact with dynamically generated products as if they were static datasets stored on disk.

```
USAGE:
    rcf2mrc.cgi [OPTIONS] <SUBCOMMAND>

OPTIONS:
    -h, --help                                          Print help information
    -n, --n-rows-max-per-cell <N_ROWS_MAX_PER_CELL>     Maximum number of rows in a MRC cell
    -V, --version                                       Print version information

SUBCOMMANDS:
    common-metadata    Creates the `_common_metadata` parquet file
    dataset            Creates the dataset directory
    help               Print this message or the help of the given subcommand(s)
    info               Creates the `catalog_info.json` file
    list               Print the page containing the list of the available catalogues (and the links)
    metadata           Creates the `_metadata` parquet file
    moc                Creates the MOC of given Norder
    partition          Creates the `parition_info.csv` file
    properties         Creates both the `properties` and the `hats.properties` file (same content)
    root               For a given catalogues, build the page containing the links to the root elements
    skymap             Creates the skymap of given Norder
    skymap-img         Creates the skymap iamge of given Norder
    sub-subdir         Creates the sub-sub-directory of given Norder and Dir number
    subdir             Creates the sub-directory of given Norder
```

Figure 2. Pure Rust CLI / CGI mimicking a static HATS product.

```
RewriteRule ^/hats/(.*)/                /cgi-bin/hats.cgi?action[root][tabname]=$2
RewriteRule ^/hats/(.*)/dataset/        /cgi-bin/hats.cgi?action[dataset][tabname]=$2
RewriteRule ^/hats/(.*)/hats.properties /cgi-bin/hats.cgi?action[properties][tabname]=$2
```

Figure 3. Examples of Apache rewrite rules.

3 - Transformation of Parquet URLs into QATSS queries

When a client requests a Parquet tile, the corresponding URL is intercepted by Apache rewrite rules and translated into a positional query handled by the *qat2s* CGI. During this process, additional metadata are retrieved from VizieR so that the generated Parquet files include VOTable metadata in their headers, in accordance with the IVOA VOParquet Note[2]. Figure 4 presents a simplified example of a rewrite rule that maps a Parquet file URL to a QATSS query, while Figure 5 shows the overall set of components involved in *qat2s* for the on-the-fly generation of Parquet tiles.

```
RewriteRule ^/hats:n=(.*)/(.*)/dataset/Norder=(.*)/Dir=.*/Npix=(.*).parquet
/cgi-bin/qat2s.cgi?file=$2&output=Parquet
&columns[cols][0]=hpx(29,DEFAULT,DEFAULT)+as+i64+as+_healpix_29
&columns[cols][1]=*
&mode[Positional][lon]=DEFAULT&mode[Positional][lat]=DEFAULT
&mode[Positional][geom][Hpx][depth]=$3&mode[Positional][geom][Hpx][hash]=$4
```

Figure 4. Simplified rewrite rule transforming a Parquet file URL into a QATSS query.

[2] https://www.ivoa.net/documents/Notes/VOParquet/

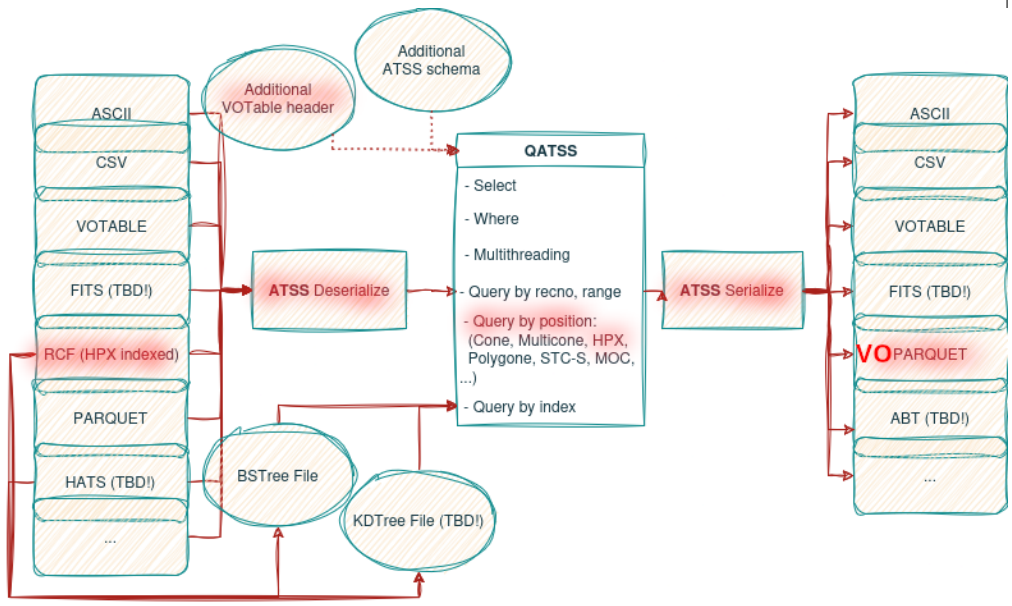


Figure 5. Components around the *qat2s* CGI used for on-the-fly Parquet files generation.

Performance and limitations

On a local network, using ZSTD compression at level 3 and Parquet row groups of 10 000 rows, the generation and download of a 404 MB Parquet tile from the Gaia DR3 main table takes approximately 20 seconds, corresponding to an effective transfer rate of about 20 MB/s. This performance could be improved by enabling multi-threaded generation or by producing a new row group while the previous one is still being transferred to the client. One of the main advantages of the Parquet format is its support for column-wise access via HTTP range requests, which allows a client to read only a subset of columns after first retrieving the file footer. This optimisation is not available for on-the-fly streamed HATS products, as the Parquet file does not exist in advance. To partially address this limitation, the LSDB/HATS interface supports a columns query parameter in Parquet file URLs, allowing column selection to be performed on the server side. Although converting data from a row-oriented format such as RCF or FITS to a column-oriented Parquet representation still requires reading all columns, the amount of data transferred over the network can be significantly reduced. In a similar way, a filters query parameter is available to apply row selection on the server.

Astronomical Data Analysis Software and Systems XXXV
*ASP Conference Series, Vol. **VOL** Compiled: Tuesday 30th June, 2026@ 01:30*
K. Polsterer, D. Bohlender, S. Gwyn, A. Hincks and P. Teuben, eds.
*© **YEAR** Astronomical Society of the Pacific*

ZZZDEBUG!236.tex
Jamal, S.|textbf
Bailer-Jones, C. A. L.

Improved quasar and galaxy classifications using the Gaia DR3

Sara Jamal,[1] and Coryn A. L. Bailer-Jones[1]

[1]*Max-Planck Institute für Astronomy, Königstuhl 17, 69117 Heidelberg, Germany;* jamal@mpia.de

Abstract. We present a performance analysis of the quasar and galaxy classifications provided by the Discrete Source Classifier (DSC) in the Gaia Data Release 3 (GDR3) and propose a new approach to combining these results in order to create higher-purity catalogues of quasars and galaxies. We reevaluate the GDR3 DSC classification performance based on source magnitudes and sky positions using a cleaner validation set and introduce a new parametric combination of the two baseline DSC classifiers that does not require any changes to the published DSC probabilities nor retraining. The results of the new classifier demonstrate a substantial enhancement in purity, albeit with a slight reduction in completeness, particularly for quasars with faint magnitudes.

1. Introduction

In the Gaia Data Release 3 (GDR3), the Discrete Source Classifier (DSC) probabilistically classifies sources in a Bayesian framework so that source can be identified as a quasar, a galaxy, a star, a white dwarf or a binary. The GDR3 DSC Combmod classifier achieved a high completeness but a low purity for quasars and galaxies due to contamination from the far larger star class. To demonstrate the variations in performance with the sources' magnitude and sky position, we compute two-dimensional (2D) representations of the completeness and the purity as a function of Galactic latitude and brightness. We also reevaluate the GDR3 DSC classifications using a cleaner validation set that excludes the Magellanic Clouds, without changes to the published probabilities. We present here the methodology and results, as discussed in Jamal & Bailer-Jones (2024).

2. Methods

DSC comprises in GDR3 three classifiers that compute posterior probabilities for five classes : quasar, galaxy, star, white dwarf, and binary star classes. Specmod is trained using the Gaia low-resolution BP/RP spectra for all the five classes, while Allosmod uses the Gaia astrometry and photometry for three classes, combining the stellar classes into one.

2.1. Probabilistic combined classifier, Combmod

The Combmod classifier is defined in GDR3 as a probabilistic combination of the posterior probabilities of Specmod and Allosmod.

$$P_k^c = \begin{cases} P_k^s \, P_k^a \, \dfrac{1}{\pi_k^s} & \text{for the quasar and galaxy classes,} \\ P_k^s \, P_k^a \, \dfrac{\pi_k^s}{(\pi_{star}^a)^2} & \text{for the stellar classes,} \end{cases} \tag{1}$$

where k, P^a, P^s, P^c and π refer to the class, the Allosmod probability, the Specmod probability, the Combmod probability and the prior function at class k, respectively. The priors $\{\pi_k^s, \pi_k^a\}$ refer to the 5-class prior used in `Specmod` and the 3-class prior used in `Allosmod`, respectively, such that $\pi^a_{\rm star}$=$\pi^s_{\rm star}$+$\pi^s_{\rm whitedwarf}$+$\pi^s_{\rm binary}$.

2.2. Parametric combined classifier, Combmod-α

We introduce a new parametric combiner from a weighted sum of the Allosmod and Specmod probabilities (no retraining of DSC models is required).

$$P_k^{c_\alpha} = \begin{cases} (1-\alpha_k) \, P_k^s + \alpha_k \, P_k^a & \text{for the quasar and galaxy classes,} \\ (1-\alpha_k) \, P_k^s + \alpha_k \, P_k^a \dfrac{\pi_k^s}{\pi_{star}^a} & \text{for the stellar classes,} \end{cases} \tag{2}$$

where k, P^a, P^s, $P^{c\alpha}$ and π refer to the class, the Allosmod probability, the Specmod probability, the Combmod-α probability and the prior function at class k, respectively. The parameters $\alpha_{\rm quasar}$, $\alpha_{\rm galaxy}$, $\alpha_{\rm star}$ are obtained via a grid search, such that the optimal values maximise the geometric score, which is defined as the geometric mean of the completeness and purity of the quasar and galaxy classes.

3. Results

The metrics computed from the confusion matrix are used evaluate performance: the completeness to the classifier's ability to identify true classes, and thee purity to assess the classifier's ability to reject false detections. It is important to note here that class imbalance is addressed in DSC in two ways, as in Bailer-Jones et al. (2019). First, the prior function is applied to the probabilities to reflect the expected fraction of each class in the data. Second, the confusion matrix is adjusted to align the test set with the expected overall class distributions. The classification metrics correspond to those computed after adjustment.

Following Hughes et al. (2022) and Jamal & Bailer-Jones (2024), classification performance is evaluated as a function of the sources' brightness and Galactic latitude . The test data is segmented onto a two-dimensional (2D) grid of 4550 bins in $|\sin b| \in [0,1]$ and $G \in [5,21]$, with coarser binning in the bright magnitude regime of $G<14$ mag. In each bin, we adjust the confusion matrix to allow for class imbalance and compute the completeness and the purity.

Metrics in Figures 1 and 2 clearly demonstrate the range of good performance and limitations such as reduced completeness and purity at fainter magnitudes. As stated in Table 1, Combmod achieves across all magnitudes average completenesses

of 92% and 95%, and average purities of 55% and 89%, for the quasar and galaxy classes, respectively. In comparison, the new classifier, Combmod-α, achieves better purities, albeit the decrease in the quasars completeness at fainter magnitudes. The most significant improvement is achieved for faint quasars at $G \geq 20$ mag, with an increase in purity from 20% to 62% despite a moderate loss in completeness. When applied to the full GDR3, the sky distributions in Jamal & Bailer-Jones (2024) show a reduced stellar contamination due to the increased purity level, as opposed to the completeness-oriented classifier Combmod.

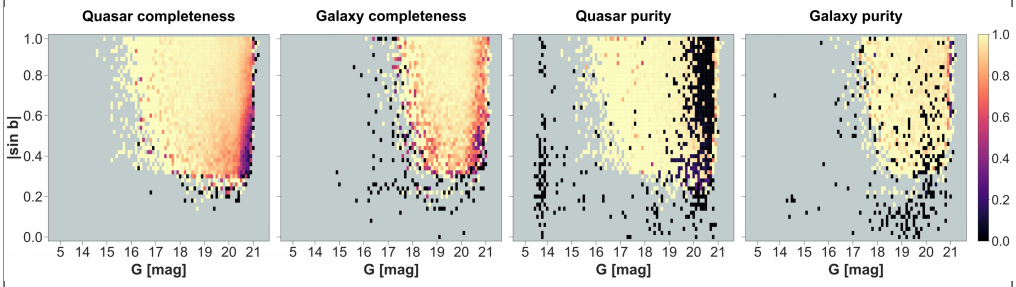


Figure 1. DSC Combmod performance in GDR3 for the quasar and galaxy classes evaluated on the test set as function of brightness and sky position. Left two panels refer to the completenesses and the right two panels to the purities. Predicted labels are obtained from the class with the highest posterior probability. Figure from Jamal & Bailer-Jones (2024)

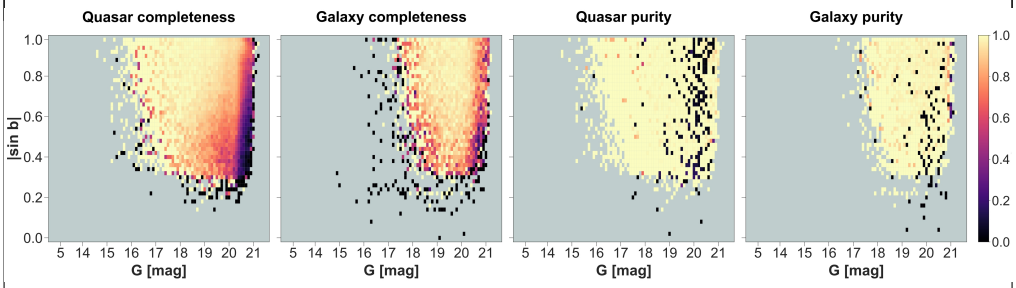


Figure 2. Same as Fig. 1 but for Combmod-α. Figure from Jamal & Bailer-Jones (2024)

Table 1. Summary of the average performance of Combmod and Combmod-α, evaluated on the cleaned test set. Values are computed from a weighted average on the 2D representations. Results from Jamal & Bailer-Jones (2024)

Magnitudes	Class	avg. completeness		avg. purity	
		Combmod	Combmod-α	Combmod	Combmod-α
All magnitudes	quasar	0.916	0.819	0.553	0.786
	galaxy	0.953	0.931	0.884	0.926
G<20 mag	quasar	0.964	0.926	0.864	0.905
	galaxy	0.967	0.949	0.918	0.958
$G \geq 20$ mag	quasar	0.864	0.706	0.196	0.616
	galaxy	0.930	0.898	0.832	0.877

4. Conclusions

We have reevaluated the performance of DSC classification results in GDR3 and introduced a new combination of DSC baseline classifiers. This new combination significantly improves the purity of extragalactic classifications, despite reduced completeness.

The new parametric combination of the baseline classifiers (Allosmod and Specmod), named Combmod-α, achieves an increase in purity by 24 and 5 percentage points but a decrease in completeness by 10 and 2 percentage points, for the quasar and galaxy classes, respectively. At magnitudes fainter than G=20 mag, the purity of the quasar class increases by 42 percentage points while the completeness decreases by only 15 percentage points. Our results demonstrate a significant improvement in purity for a modest loss of completeness for faint quasars, based solely on a new combination of the published GDR3 DSC probabilities.

Acknowledgments. This work was funded in part by the DLR (German space agency) via grant 50 QG 2102 . This work has made use of data from the European Space Agency (ESA) mission Gaia [1], processed by the Gaia Data Processing and Analysis Consortium (DPAC)[2]. Funding for the DPAC has been provided by national institutions, in particular the institutions participating in the Gaia Multilateral Agreement.

References

Bailer-Jones, C. A. L., Fouesneau, M., & Andrae, R. 2019, MNRAS, 490, 5615
Hughes, A. C. N., Bailer-Jones, C. A. L., & Jamal, S. 2022, A&A, 668, A99
Jamal, S., & Bailer-Jones, C. A. L. 2024, A&A, 687, A258

[1] https://www.cosmos.esa.int/gaia
[2] https://www.cosmos.esa.int/web/gaia/dpac/consortium

Astronomical Data Analysis Software and Systems XXXV
*ASP Conference Series, Vol. **VOL** Compiled: Tuesday 30th June, 2026@ 01:30*
K. Polsterer, D. Bohlender, S. Gwyn, A. Hincks and P. Teuben, eds.
*© **YEAR** Astronomical Society of the Pacific*

ZZZDEBUG!237.tex
Gurgul, A.|textbf
Wyrzykowski, Ł.
Kotysz, K.
Mikołajczyk, J. P.
Zieliński, P.
archives!individual!GAIA
catalogues!services!Gaia
Archive Core Systems
astronomy!photometry
astronomy!astrometry
images!processing!photomet
software!image
processing
software!tools!CCDPhot
algorithm!SExtractor
algorithm!CPCS
organisations!EASST
BHTOM
GAIA! ascl:1403.024
AMPEL! ascl:2005.015
ATLAS! ascl:1911.013
SWIFT! ascl:1805.020

BHTOM.space - tool for automatic processing of time-domain data

Agnieszka Gurgul,[1] Łukasz Wyrzykowski,[2,3] Krzysztof Kotysz,[2,4] Przemysław J. Mikołajczyk,[3,4] and Paweł Zieliński[1]

[1]*Institute of Astronomy, Nicolaus Copernicus University in Toruń, Toruń, Poland;* agn.gurgul@umk.pl

[2]*Astronomical Observatory, University of Warsaw, Warsaw, Poland*

[3]*National Centre of Nuclear Research, Warsaw, Poland*

[4]*Astronomical Institute, University of Wrocław, Wrocław, Poland*

Abstract. BHTOM.space is a powerful platform designed to coordinate the photometric observations from a heterogeneous global network of small- to medium-sized telescopes, including both professional and amateur observatories. Built on a modular Django/Python backend with PostgreSQL and RESTful APIs, it supports automated ingestion and processing of photometric data (FITS, CSV), real-time cross-matching, and target prioritisation, as well as querying numerous time-domain photometric archives. Through this system, we have collected and processed hundreds of thousands of photometric data points with the CCDPhot pipeline, all fully calibrated and standardised with CPCS and ready for scientific use. The BHTOM.space provides researchers with access to high-quality, long-term datasets for studying transient objects and variable stars. We invite telescope owners and astronomers working with time-domain data to contribute to and explore our ever-growing database, and to leverage our network for future observations.

1. Introduction

Time-domain astronomy is increasingly driven by large-scale surveys such as Gaia, ZTF, and the forthcoming LSST, which generate alerts and photometric time series for millions of variable and transient sources. The scientific exploitation of these discoveries - including supernovae and other transients, quasar and blazar variability, gravitational microlensing, young stellar objects, eclipsing binaries, planetary transits, and stellar variability - requires extensive ground-based follow-up. In many cases, flexible cadence and long temporal baselines are more important than large aperture, placing small- and medium-sized telescopes in a key supporting role.

A large population of telescopes with apertures ranging from a few centimetres to approximately 2.5 m is distributed globally and operated by professional observatories, universities, and experienced amateur astronomers. Although individual facilities often have limited fields-of-view and heterogeneous instrumentation, their geographic distribution provides resilience against local weather and enables long-term or near-continuous monitoring. When observations from such facilities can be efficiently co-

ordinated and photometrically standardised, they provide high scientific value complementary to wide-field survey data.

Modern time-domain discovery is increasingly mediated through alert brokers, which aggregate, classify, and distribute alerts originating from survey pipelines. Brokers such as ANTARES, AMPEL, and others provide enriched alert streams that can serve as inputs for downstream follow-up systems. In addition, curated sources of transient targets, including the Transient Name Server (TNS) and survey-specific alert feeds, represent important entry points for photometric monitoring. A flexible follow-up infrastructure must therefore be capable of ingesting targets from multiple heterogeneous sources, ranging from real-time broker streams to manually curated or externally triggered target lists.

The BHTOM[1] (Black Hole Target Observation Manager) network was originally developed to support rapid photometric follow-up of alerts from ESA's Gaia mission. Early implementations were deployed as part of the Gaia Science Alerts infrastructure, with initial operations beginning in 2013 via the Cambridge Calibration Photometric Server (CPCS). These systems provided automated photometric processing and validation for transient alerts, enabling coordinated follow-up by a distributed telescope network (Wyrzykowski et al. 2012; Zieliński et al. 2019; Hodgkin et al. 2021; Mikolajczyk et al. 2025).

Since 2019, BHTOM has been built upon the Las Cumbres Observatory Target and Observation Manager (TOM) Toolkit, which provides a modular framework for target and observation management (Street et al. 2018). BHTOM extends the core TOM functionality to operate on a highly heterogeneous network of telescopes by introducing automated ingestion of photometric data from arbitrary facilities, processing of FITS images and tabular datasets, and standardised photometric calibration across instruments. The system also allows users to query multiple time-domain archives, providing uniform access to historical and contemporaneous photometric data.

In this paper, we present `BHTOM.space` as a scalable platform that bridges discovery surveys, alert brokers, and globally distributed follow-up resources. By unifying target ingestion, data processing and photometric standardisation BHTOM enables small- and medium-sized telescopes to collectively deliver high-quality time-domain datasets for a broad range of astrophysical applications.

2. BHTOM.space platform

BHTOM is a web-based system built on the Django framework. It links large-scale survey facilities (e.g. Gaia, ZTF, ATLAS) and alert infrastructures with a heterogeneous, global network of small- and medium-sized telescopes. The platform supports filtering (by magnitude, position, Sun separation, observing priority, etc.) and visualisation of target lists, as well as interactive inspection of photometric data from ongoing follow-up observations and a wide range of publicly available time-domain archives, from high energy UV (GALEX, Swift), through the optical domain (e.g. Gaia, ZTF, ASASSN, CRTS, OGLE, LINEAR, SDSS), up to low energectic radio waves (LOFAR, FIRST). The system also provides a comprehensive set of RESTful API endpoints, enabling seamless integration with external services and adaptation to diverse science use cases.

[1] Black Hole TOM: https://bhtom.space

The operational workflow of BHTOM can be summarised in the following stages: (1) Ingestion of alerts and targets from external surveys, brokers, and catalogues; (2) Selection and triggering of observations at suitable facilities, using both automated and manual scheduling; (3) Automated processing of incoming data using the in-house `CCDPhot` photometric and astrometric pipeline; (4) Photometric calibration, standardisation - including colour term correction - and the generation of multi-band, long-term light curves.

This end-to-end workflow enables efficient, reproducible, and science-ready photometric follow-up without the need for proposal-driven access or manual coordination, thereby lowering the barrier for participation by both professional and non-professional observatories.

3. CCDPhot image processing pipeline

`CCDPhot` is an image processing and photometry engine within the BHTOM platform, designed to provide reliable and robust handling of calibrated FITS images from a highly diverse network of telescopes. It currently supports data from over 160 facilities equipped with various cameras, optics and filter sets, processing an average of over 50,000 observations per month. The pipeline performs astrometric calibration, source detection, circular and kron-like aperture and PSF photometry in a fully automated manner, enabling uniform processing of data acquired under diverse observational conditions and supplying multiple data products back to the BHTOM system.

To ensure interoperability, `CCDPhot` performs standardisation of FITS headers before processing, allowing images originating from a wide range of telescope control systems to be handled consistently. Internally, the pipeline builds upon well-established and extensively tested astronomical software packages, including `DAOPHOT` ((Stetson 1987)) for PSF photometry, `SExtractor` ((Bertin & Arnouts 1996)) for source detection, and `SCAMP` ((Bertin 2006)) for astrometric solutions.

The primary outputs of `CCDPhot` are calibrated astrometric positions and instrumental photometry. These measurements are standardised using the Gaia Synthetic Photometry catalogue, which provides an uniform photometric calibration across different instruments and filter systems. This enables the generation of consistent multi-band photometric time series that are suitable for further scientific analysis.

4. Global telescope network

Within the OPTICON-RadioNet Pilot (2021–2025) and ACME (2025–2028) EU programmes, BHTOM.space coordinates observations from a distributed network of approximately 160 small- and medium-sized telescopes (0.05–2.5 m) across Europe and beyond (see Figure 2). Many of these facilities, including university and amateur observatories that are otherwise underutilised, deliver scientifically valuable data when integrated into a coordinated network.

Additionally, through the `EASST.eu` foundation, experienced amateur astronomers are actively engaged in professional research programmes, contributing calibrated photometric data to ongoing scientific campaigns. With the number of registered users approaching 700, the BHTOM network continues to grow as an open and collaborative

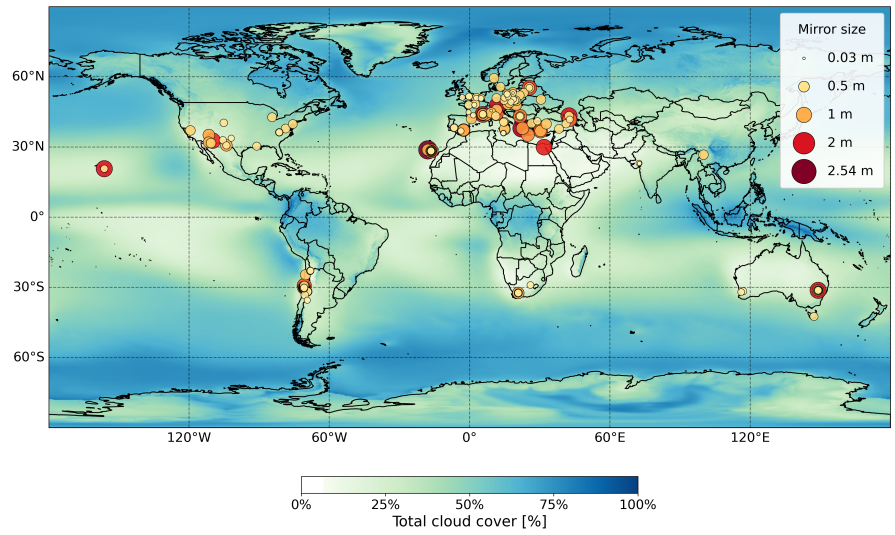


Figure 1. Geographical distribution of the global telescope network participating in BHTOM.space overplotted on the total average cloud cover ((Brun et al. 2022)).

platform, welcoming new observatories and users operating both manual and robotic facilities.

Acknowledgments. This project has received funding from the European Union's Horizon Europe Research and Innovation programme ACME under grant No 101131928 (2024-2028) and from the Polish National Science Centre DAINA grant No 2024/52/L/ST9/00210.

References

Bertin, E. 2006, in Astronomical Data Analysis Software and Systems XV, edited by C. Gabriel, C. Arviset, D. Ponz, & S. Enrique, vol. 351 of Astronomical Society of the Pacific Conference Series, 112

Bertin, E., & Arnouts, S. 1996, A&AS, 117, 393

Brun, P., Zimmermann, N. E., Hari, C., Pellissier, L., & Karger, D. N. 2022, Earth System Science Data, 14, 5573

Hodgkin, S. T., et al. 2021, A&A, 652, A76. eprint: 2106.01394

Mikolajczyk, P. J., Zieliński, P., Wyrzykowski, L., Krawczyk, A., & Kotysz, K. 2025, in Revista Mexicana de Astronomia y Astrofisica Conference Series, vol. 59 of Revista Mexicana de Astronomia y Astrofisica Conference Series, 167

Stetson, P. B. 1987, PASP, 99, 191

Street, R. A., Bowman, M., Saunders, E. S., & Boroson, T. 2018, in Software and Cyberinfrastructure for Astronomy V, edited by J. C. Guzman, & J. Ibsen, vol. 10707 of Society of Photo-Optical Instrumentation Engineers (SPIE) Conference Series, 1070711. eprint: 1806.09557

Wyrzykowski, Ł., Hodgkin, S., Blogorodnova, N., Koposov, S., & Burgon, R. 2012, in 2nd Gaia Follow-up Network for Solar System Objects, 21. eprint: 1210.5007

Zieliński, P., Wyrzykowski, Ł., Rybicki, K., Kołaczkowski, Z., Bruś, P., & Mikołajczyk, P. 2019, Contributions of the Astronomical Observatory Skalnate Pleso, 49, 125

Astronomical Data Analysis Software and Systems XXXV
*ASP Conference Series, Vol. **VOL** Compiled: Tuesday 30th June, 2026@ 01:30*
K. Polsterer, D. Bohlender, S. Gwyn, A. Hincks and P. Teuben, eds.
*© **YEAR** Astronomical Society of the Pacific*

ZZZDEBUG!239.tex
Kretlow, M.|textbf
Ortiz, J.-L.
astronomy!solar system
FOOBAR! ascl:1101.010

CORA: A Community Platform for Stellar Occultations and Small Solar System Bodies Research

Mike Kretlow[1] and Jose-Luis Ortiz[2]

[1]*Deutsches Zentrum für Astrophysik (DZA), 02826 Görlitz, Germany;* mike.kretlow@dzastro.de

[2]*Instituto de Astrofísica de Andalucía (IAA-CSIC), Granada, Spain*

Abstract. Stellar occultations provide unique constraints on the sizes, shapes, and orbits of small Solar System bodies, but their scientific exploitation requires extensive coordination between prediction pipelines, observers, and data analysis workflows. CORA (Collaborative Occultation Resources and Archive) is an online, community-driven platform designed to support the full lifecycle of stellar occultation campaigns, from prediction and visualization to observation collection and preliminary analysis. CORA emphasizes open access, reproducibility, and collaboration between professional and amateur astronomers.

1. Introduction

Stellar occultations are among the most powerful techniques for directly measuring the physical properties of small Solar System bodies (SSBs), including asteroids, Centaurs, and trans-Neptunian objects. Multi-chord occultation observations can yield kilometer-scale size estimates, shape constraints, and accurate astrometry (Kaminski et al. 2023) that is difficult to achieve by other observational methods. Furthermore, the environment of the occulting object is also probed during a stellar occultation, with the potential to reveal satellites (Gault et al. 2022), rings (Braga-Ribas et al. 2014) or an atmosphere down to the sub-microbar pressure level (Hubbard et al. 1988). Although not the main application of stellar occultations by SSBs (except for occultations of our Moon), the target star itself may be the object to be studied (Kretlow et al. 2023) or the subject of a serendipitous discovery of its binary/multiple nature (Andersson 2022).

At the same time, occultation science poses significant organizational and technical challenges. Reliable predictions require up-to-date orbits, observations are typically carried out by geographically distributed observers, and the resulting data must be aggregated and interpreted on an event-by-event basis. Historically, these activities have been handled through a collection of partially disconnected tools, mailing lists, and ad-hoc data exchanges.

CORA was developed to address these challenges by providing a unified, web-based platform that supports coordination, data sharing, and analysis across the occultation community.

2. Design Goals and Scope

From an infrastructure perspective, CORA aims to provide a persistent and reproducible framework for occultation predictions, observation archiving, and analysis, and is designed in accordance with the FAIR principles (Findable, Accessible, Interoperable, Reusable). Rather than serving as a single-purpose tool, it is conceived as a modular platform that can evolve alongside improvements in orbits, stellar catalogs, and analysis techniques.

From a community perspective, CORA is designed to support collaboration both within and between data creators (observers) and data users such as professional astronomers. Occultation campaigns frequently rely on small and mid-sized telescopes distributed worldwide, making the participation of experienced amateur observers and citizen scientists, including users of smart telescopes, an essential component. CORA provides shared access to predictions, visualizations, and results, thereby lowering the barrier to coordinated observing campaigns.

The platform currently focuses on stellar occultations by small Solar System objects, including main-belt asteroids, near-Earth objects, Centaurs, and trans-Neptunian objects.

3. System Architecture and Data Flow

CORA is implemented as a web-based platform with a backend architecture that separates computation, data storage, and user interaction. At a conceptual level, the system consists of three main components: prediction services, data management, and visualization and analysis tools.

Occultation predictions are generated using up-to-date orbital solutions and stellar catalogs, with configurable filtering criteria tailored to different classes of objects. Predicted events are stored and made accessible through interactive visualization tools, including Earth-based ground-track maps.

Observational data are ingested either through standardized submission forms or via external feeds such as SODIS (Stellar Occultation Data Input System) operated by IOTA/ES (Andersson et al. 2023) and the OCCULT program (Herald et al. 2020) database. The platform associates individual observations with predicted events, enabling event-centric aggregation of heterogeneous data sets.

4. User Workflows and Community Interaction

A typical CORA workflow begins with the identification of candidate occultation events. Users can explore predictions through interactive maps and tables, filtering events by object type, stellar magnitude, geographic visibility, and temporal constraints.

Once observations are available for an event (ingested as described in the previous section), CORA provides preliminary visualization products such as station maps and chord plots, together with filtering options for accessing and inspecting the observational data.

Importantly, CORA acts as a shared archive. Even single-chord or negative observations are preserved, recognizing their value for orbit refinement and statistical studies.

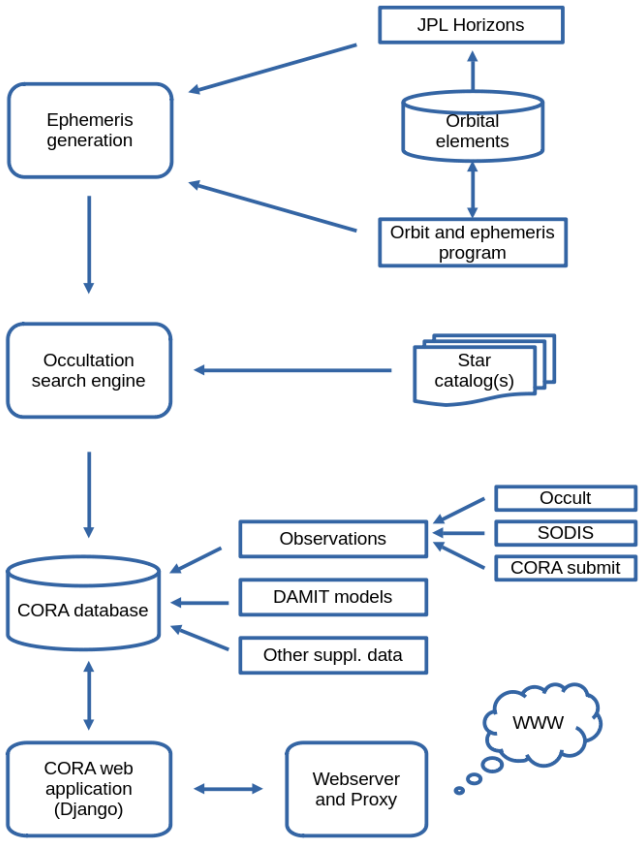


Figure 1. CORA system architecture and data flow, showing how occultation predictions and observational reports from multiple sources are ingested, processed, and stored in a central database. These data are accessed by the CORA web portal to provide visualization, coordination, and preliminary analysis tools to end users.

5. Current Status and Future Development

CORA (https://smallbodies.org/cora) is currently operational as an actively maintained platform and is used in ongoing occultation campaigns. Development to date has focused on robustness, usability, and integration of diverse data sources.

Looking ahead, CORA is being developed with scalability in mind, particularly in anticipation of the Rubin Observatory era. The dramatic increase in the number of known small bodies will lead to a substantial growth in predicted occultation events. Supporting efficient filtering, prioritization, and community coordination at this scale is a central design objective.

Planned developments include tighter integration with automatic orbit improvement workflows, enhanced support for observer collaboration and campaign planning, automated observations and scheduling support, and expanded analysis tools for multi-event and population-level studies.

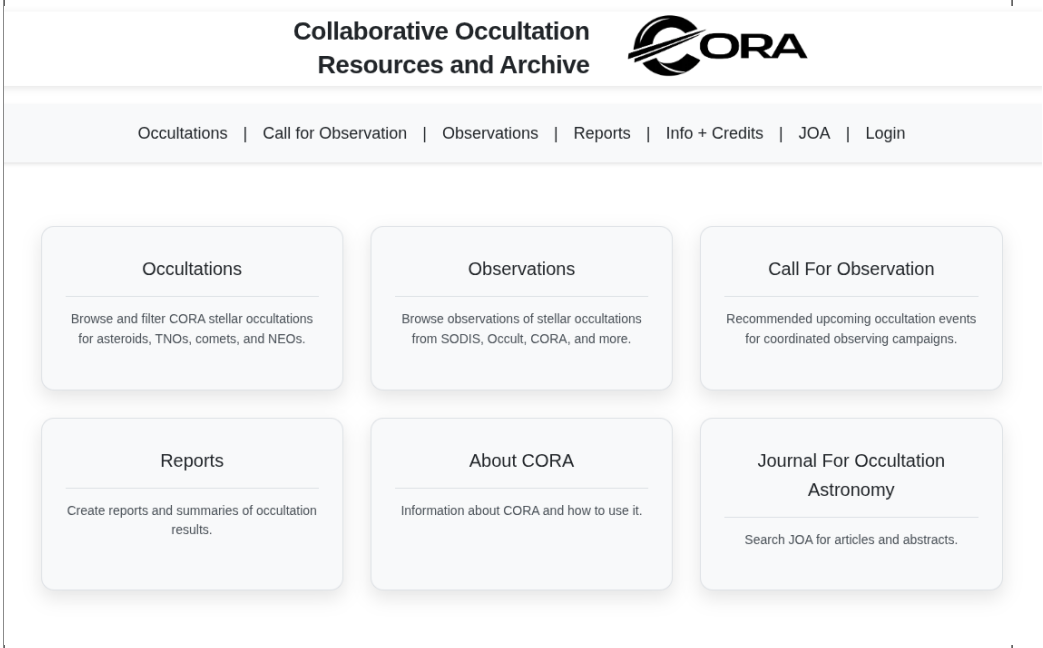


Figure 2. Screenshot of the CORA dashboard, showing the web-based user interface used to explore predicted occultation events and associated observational data. The dashboard provides interactive access to event lists, maps, and status information, supporting coordination, data inspection, and community-driven observing campaigns.

6. Conclusions

CORA provides a unified, community-oriented platform for stellar occultation science, addressing both the technical and organizational challenges inherent to this observational technique. By combining prediction, coordination, data archiving, and preliminary analysis within a single infrastructure, CORA lowers barriers to participation and supports reproducible, collaborative research.

As occultation science enters a data-rich era driven by large surveys and improved catalogs, platforms like CORA will play an increasingly important role in enabling effective community engagement and maximizing scientific return.

References

Andersson, S. 2022, Journal for Occultation Astronomy, 12, 7
Andersson, S., Guhl, K., & Tunsch, E. 2023, Journal for Occultation Astronomy, 13, 22
Braga-Ribas, F., Sicardy, B., Ortiz, J. L., et al. 2014, Nature, 508, 72
Gault, D., Nosworthy, P., Nolthenius, R., et al. 2022, Minor Planet Bulletin, 49, 3
Herald, D., Gault, D., Anderson, R., et al. 2020, MNRAS, 499, 4570
Hubbard, W. B., Hunten, D. M., Dieters, S. W., et al. 1988, Nature, 336, 452
Kaminski, K., Weber, C., Marciniak, A., et al. 2023, PASP, 135, 025001
Kretlow, M., Ortiz, J. L., & Schnabel, C. 2023, Journal for Occultation Astronomy, 13, 3

Astronomical Data Analysis Software and Systems XXXV
*ASP Conference Series, Vol. **VOL** Compiled: Tuesday 30th June, 2026@ 01:30*
K. Polsterer, D. Bohlender, S. Gwyn, A. Hincks and P. Teuben, eds.
*© **YEAR** Astronomical Society of the Pacific*

ZZZDEBUG!240.tex
Baumann, M.|textbf
Fernique, P.
Boch, T.
data!Hierarchical Progressive Surveys (HiPS)
applications!Aladin Lite
libraries!WebGL
observatories!ground-based
web!application
libraries!HEALPix
Aladin! ascl:1112.019
HEALPix! ascl:1107.018
Spectra! ascl:1701.003

Implementation of HiPS3D in Aladin Lite within the SRCNet framework.

Matthieu Baumann,[1] Pierre Fernique,[2] and Thomas Boch[2]

[1]*Université de Strasbourg, CNRS, Observatoire astronomique de Strasbourg, UMR 7550, 11 rue de l'Université, F-67000 Strasbourg, France;*
matthieu.baumann@astro.unistra.fr

[2]*Université de Strasbourg, CNRS, Observatoire astronomique de Strasbourg, UMR 7550, 11 rue de l'Université, F-67000 Strasbourg, France*

Abstract. We will present the implementation of the **HiPS3D** extension in Aladin Lite, enabling efficient visualization and exploration of large cube datasets directly in the browser. This work, carried out within the SRCNet framework, demonstrates how HiPS3D technology can support interactive navigation through GBs-TBs of spectral cube data. This article will focus on the technical details of the integration, its performance, and the potential of HiPS3D for next-generation astronomical data exploration tools.

1. What is the HiPS Format ?

HiPS (Fernique et al. 2017), short for Hierarchical Progressive Survey, is an IVOA standard designed to make sky surveys accessible and viewable regardless of the survey's size, the network quality, or the computing resources available to the astronomer.

HiPS makes Aladin Lite to only request for HEALPix (Górski et al. 2005) tile images located in the user's screen. Therefore, the more you zoom in the survey, the more details you see.

2. The Challenge of New Cubic Data: HiPS Must Evolve

New missions such as SKA will provide vast quantities of cubic radio observational data on the 100 TB to PB scale (Salgado et al. 2024). A cubic extension to the HiPS standard, named HiPS3D, is being proposed to enable navigation within a cubic mosaic of observations as part of the IVOA HiPS 2.0 standard.

3. Implementation Notes in Aladin Lite

3.1. Tile Selection and Data Access

The implementation of HiPS3D in Aladin Lite (Baumann et al. 2022) follows the same general principles as the standard HiPS framework. At any given time, only the cubic

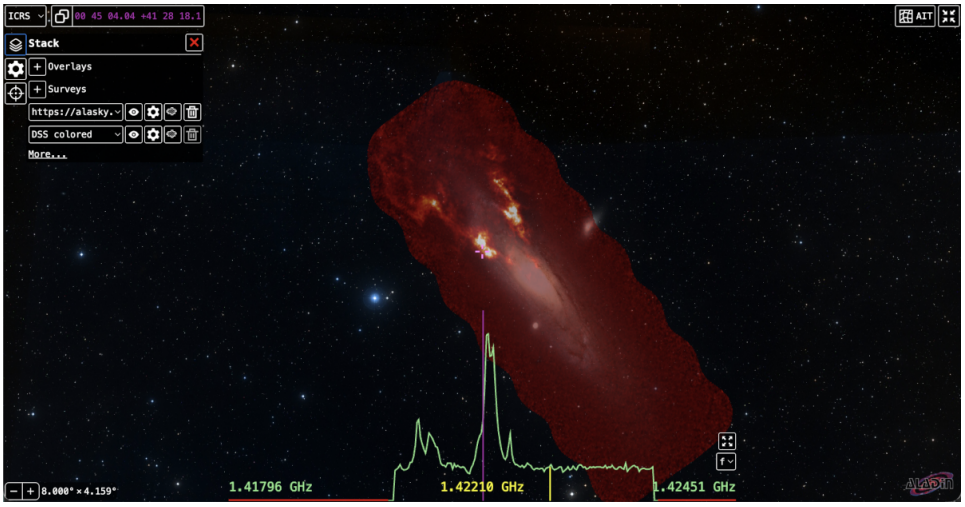


Figure 1. Display of the HiPS3D of the Local Group L-Band Survey (LGLBS) on top the classic Digital Sky Survey HiPS

tiles intersecting the current user view are requested from the server. This selection is performed both in the spatial domain and along the spectral axis.

In addition to the tiles required for the visible volume rendering, extra cubic tiles are fetched to allow continuous spectral inspection beneath the reticle. This ensures that spectra can be displayed smoothly, even when the corresponding frequency slices are not part of the currently rendered volume.

Each cubic tile is stored as an individual file on the server. Both FITS and standard image formats e.g. PNG or JPEG are supported.

- FITS tiles contain volumetric data that can be directly interpreted as voxel values.
- Image-based tiles are composed of a stack of two-dimensional slices.

3.2. GPU Representation and WebGL Integration

Cubic tiles are stored directly in GPU memory as `WebGL2 Texture3D` objects. For FITS tiles, voxel data are mapped directly into the corresponding three-dimensional texture without intermediate conversion. For PNG or JPEG tiles, the individual image slices are reassembled on the client side to construct the `Texture3D` object.

The core of the HiPS3D rendering pipeline has been implemented in Rust and compiled to WebAssembly. The `wasm-bindgen` crate is used to provide bindings between Rust and the WebGL2 API, allowing GPU resources to be created and managed efficiently from Rust code while remaining fully interoperable with the JavaScript environment.

3.3. GPU Tile Cache Management

To control GPU memory usage and maintain interactive performance, a fixed-size cache of cubic tiles is maintained on the GPU. The cache can hold up to 512 tiles simultaneously. When its limit are reached, newly requested tiles replace the oldest ones using a first-in, first-out eviction strategy.

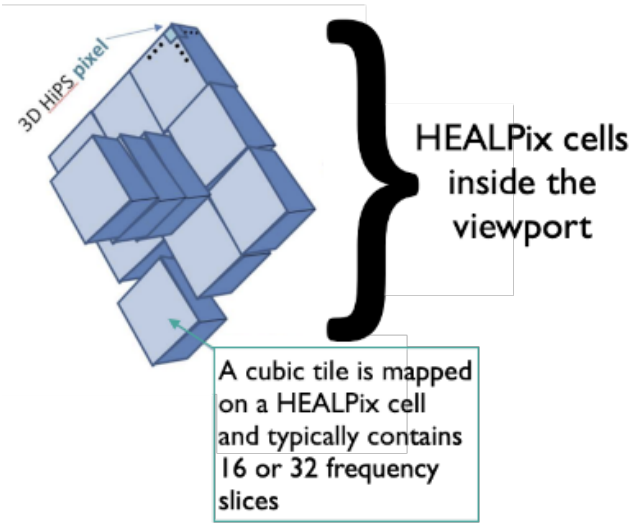


Figure 2. Cubic tile inside the view to download

This caching mechanism reduces unnecessary data transfers between the CPU and GPU while ensuring that tiles relevant to recent user interactions remain readily available for rendering.

3.4. Spectral Navigation and User Interaction

A dedicated spectral display widget has been implemented in vanilla JavaScript to support interactive exploration of the spectral dimension. Users can navigate through frequency space by directly grabbing and moving the displayed spectrum. A selector allows the user to change the displayed unit between frequency, wavelength or velocity.

As the user interacts with the spectral widget, additional cubic tiles corresponding to the newly displayed frequency ranges are retrieved on the fly from the server.

3.5. Implementation Details

Further technical details, implementation discussions, and code examples can be found in the corresponding Aladin Lite pull request: `https://github.com/cds-astro/aladin-lite/pull/314`.

4. Conclusion

The upcoming cubic data from the SKA mission motivates the ongoing development of HiPS3D and its implementation in Aladin Lite. This approach provides an efficient way to explore and visualize complex 3D astronomical data directly in the browser. Using hierarchical structures and GPU-based rendering, it enables interactive navigation of spectral cubes. While still a work in progress, HiPS3D illustrates how existing tools can evolve to meet the challenges of nextgeneration cubic surveys.

Acknowledgments. The new HiPS3D support implementation within Aladin Lite has been a work part of the SRCNet mango team during the PI27.

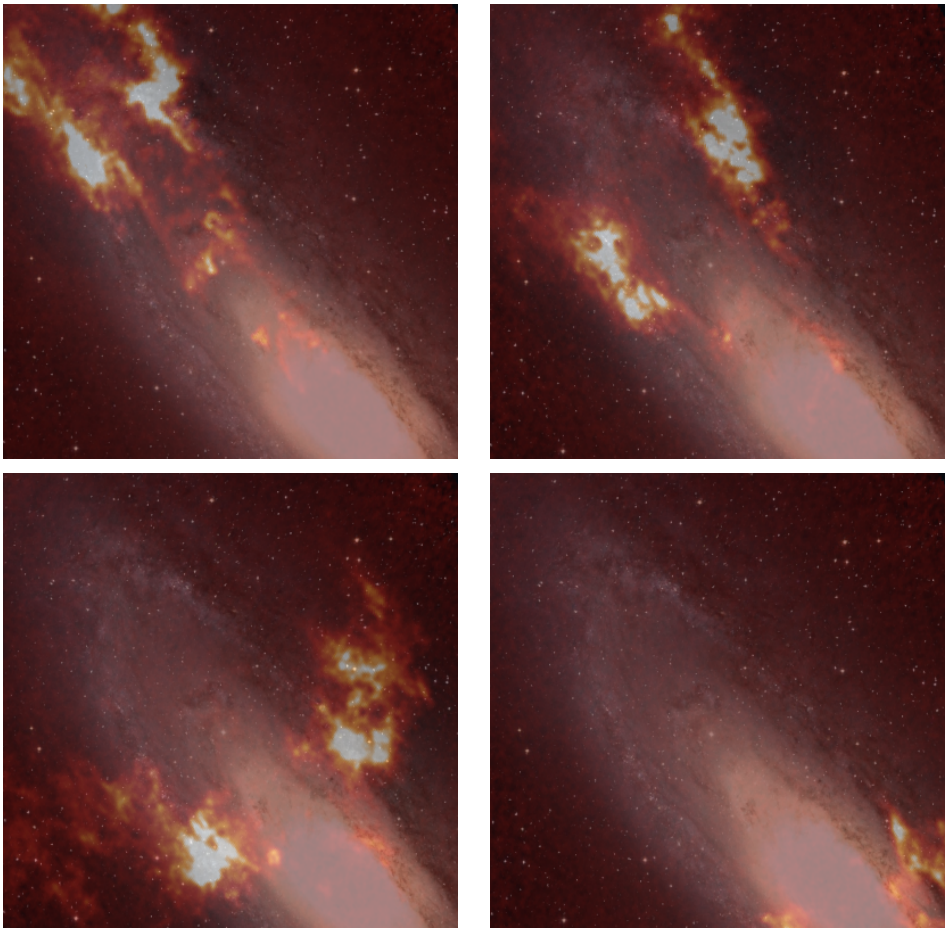


Figure 3. From top to bottom, left-right order. HiPS displayed at 1.42080 Ghz, 1.42120 Ghz, 1.42160 Ghz and 1.42200 Ghz

References

Baumann, M., Boch, T., Pineau, F.-X., Fernique, P., Bot, C., & Allen, M. 2022, in Astronomical Data Analysis Software and Systems XXX, edited by J. E. Ruiz, F. Pierfedereci, & P. Teuben, vol. 532 of Astronomical Society of the Pacific Conference Series, 7

Fernique, P., Allen, M., Boch, T., Donaldson, T., Durand, D., Ebisawa, K., Michel, L., Salgado, J., & Stoehr, F. 2017, HiPS - Hierarchical Progressive Survey Version 1.0, IVOA Recommendation 19 May 2017. eprint: 1708.09704

Górski, K. M., Hivon, E., Banday, A. J., Wandelt, B. D., Hansen, F. K., Reinecke, M., & Bartelmann, M. 2005, The Astrophysical Journal, 622, 759. URL `https://doi.org/10.1086/427976`

Salgado, J., Kitaeff, S., Bolton, R., Chrysostomou, A., Swinbank, J., Vilotte, J., Goliath, S., Gaudet, S., Barbosa, D., Yates, J., Ferrari, C., Beswick, R., Wadadekar, Y., Bartolini, M., & Rees, N. 2024, in Astromical Data Analysis Software and Systems XXXI, edited by B. V. Hugo, R. Van Rooyen, & O. M. Smirnov, vol. 535 of Astronomical Society of the Pacific Conference Series, 399

Astronomical Data Analysis Software and Systems XXXV
*ASP Conference Series, Vol. **VOL** Compiled: Tuesday 30th June, 2026@ 01:30*
K. Polsterer, D. Bohlender, S. Gwyn, A. Hincks and P. Teuben, eds.
*© **YEAR** Astronomical Society of the Pacific*

ZZZDEBUG!241.tex
Paquin, J.|textbf
Lockhart, K.
Blanco-Cuaresma, S.
Accomazzi, A.
Kurtz, M.
Henneken, E.
Grezes, F.
Allen, T.
Grant, C.
Hostetler, T.
Templeton, M.
Chen, S.
Koch, J.
Jacovich, T.
Chivvis, D.
de Macedo Alves, F.
Astrophysics Data System (ADS)
FOOBAR! ascl:1101.010

Identifying Instruments in the ADS/SciX Corpus with LLM Agents

Jean-Claude Paquin,[1] Kelly E. Lockhart,[1] Sergi Blanco-Cuaresma,[1,2] Alberto Accomazzi,[1] Michael J. Kurtz,[1] Edwin A. Henneken,[1] Felix Grezes,[1] Thomas Allen,[1] Carolyn S. Grant,[1] Timothy W. Hostetler,[1] Matthew R. Templeton,[1] Shinyi Chen,[1] Jennifer Koch,[1] Taylor Jacovich,[1] Daniel Chivvis,[1] Fernanda de Macedo Alves,[1] Jennifer Bartlett,[1] Mugdha Polimera,[1] and The ADS/SciX Team

[1]*Harvard-Smithsonian Center for Astrophysics, Cambridge, MA, USA;* jean-claude.paquin@cfa.harvard.edu

[2]*Faculty of Psychology, UniDistance Suisse, Brig, Switzerland*

Abstract. Bibliographies are a core tool used by observatories to evaluate the impact of their facilities and instruments. Yet, identifying and classifying papers referencing specific instruments is usually a manual, time-intensive task. We developed a large language model (LLM)-augmented pipeline to automatically construct a comprehensive list of instruments referenced across the full astronomy corpus of the Astrophysics Data System (ADS/SciX), roughly 3 million records in size. By grounding LLM agents with web search, we increased the number of true-positive text-to-instrument associations. What would have taken multiple weeks of focused work by a single human curator was accomplished in hours, and can now be run incrementally on new additions to the corpus to dynamically identify novel instruments as they appear in the literature.

1. Introduction

The Astrophysics Data System (ADS/SciX) serves the global astronomy research community by providing comprehensive search functionality over the literature of astronomy and related fields(Bartlett et al. 2025). Users frequently need to search for papers that reference specific instruments, telescopes, and facilities. Observatories, for example, traditionally maintain their own bibliographies through labor-intensive manual searches, often using ADS/SciX, and paper-by-paper review(Observatory Bibliographers Collaboration et al. 2024). Such bibliographies help observatories evaluate the scientific impact of their infrastructure (Observatory Bibliographers Collaboration et al. 2024; Mayernik et al. 2017). ADS/SciX provided instrument associations for these use-cases in the past(Grant et al. 2000). However, instrument data in ADS/SciX is incomplete and infrequently updated. Manual curation of instrument mentions across millions of papers is prohibitively time consuming (Accomazzi et al. 2012) and cannot scale with the continuous growth of ADS/SciX's corpus.

Recent advances in large language models (LLMs)(Li et al. 2025) offer new opportunities to automate information extraction tasks at scale. However, LLMs face challenges with hallucinations(Tonmoy et al. 2024) and limited knowledge of specialized domain content(Li et al. 2025), particularly for lesser-known instruments. These

challenges can be mitigated via Retrieval-Augmented Generation (RAG)(Tonmoy et al. 2024), such as web or document search, to ground results.

This paper describes our multi-stage pipeline for automatically identifying astronomical instruments mentioned across the complete ADS/SciX corpus. We use traditional natural language processing techniques for candidate extraction, aggregating results to ensure licensing compliance and reduce the cost of further steps. Then, with search-grounded, LLM-based entity resolution, we produce a list of instruments. Note that our method specifically does not link papers to instruments; we instead rely on other systems to perform disambiguation per-paper.

2. Methods

Our instrument identification pipeline consists of two main phases: candidate *generation* and *merging*. The generation phase uses traditional text processing, web search, and LLM-based filtering to generate possible candidates for n-grams (contiguous sequences of n tokens, e.g., words or subword units) selected from across the corpus. The merging phase takes generated candidates and merges them to eliminate redundant candidates.

2.1. Candidate Generation

2.1.1. N-gram Selection

The *Candidate Generation* phase begins by selecting papers from the ADS/SciX corpus that are likely to mention astronomical instruments. We queried ADS/SciX for all papers containing the term "telescope" in their title, abstract, or full text body. Of the approximately 1.5 million astronomy papers with full text in ADS/SciX, this initial filter yielded approximately 600,000 papers. While this approach may miss papers discussing instruments without using the specific term "telescope," it provides a manageable subset highly enriched for relevant mentions.

Next, to identify potential instrument names, we extracted n-grams from the full text of all selected papers. First, we normalized and tokenized the text using the following steps: **1.** Unicode normalization (NFKD form); **2.** Removal of punctuation; **3.** Conversion to lowercase; **4.** Reduction of multi-character whitespace sequences to single spaces; **5.** Splitting on spaces into tokens.

We defined a set of trigger words commonly associated with astronomical instruments: "telescope," "spectrometer," "explorer," "satellite," "spectrograph," "polarimeter," "laboratory," "observatory," "imager," "survey," "observer," and "array."

When a trigger word was identified in the text, we extracted 6-grams, 5-grams, 4-grams, and 3-grams ending with that trigger word. This captures common instrument naming patterns such as "the very large telescope" or "the hubble space telescope."

To reduce noise and focus on commonly mentioned instruments, we applied frequency based filtering: only n-grams that appeared at least 20 times across the corpus were selected. Additionally, we limited candidates to n-grams beginning with the article "the," as this pattern is characteristic of formal instrument names. This filtering reduced our candidate set to approximately 11,000 unique n-grams.

2.1.2. N-gram Filtering

For each candidate n-gram, we performed a web search using the Brave Search API to gather contextual information. We appended the word "astronomy" to each n-gram in the search query to bias results toward astronomical content. We collected the top K=2 search result descriptions for each n-gram to provide grounding information for subsequent LLM classification.

We used *Qwen3 235B A22B 2507*(Qwen Team 2025) to classify whether each n-gram refers to a real astronomical telescope, instrument, array, or facility. The model was provided with the n-gram and its associated web search results as context. For each n-gram, we requested the following fields: `is_real_telescope`, a boolean indicating whether the n-gram refers to a real instrument; `canonical_name`, the official name used in technical reports and documentation; `alternative_names`, other commonly used names for the instrument; `acronyms`, standard acronyms associated with the instrument.

To account for variability in LLM responses, we queried the model 3 times for each n-gram and selected the mode response; responses with no identifiable mode were filtered out. This initial mapping stage reduced our candidate set to approximately 1,700 identified canonical names.

2.2. Candidate Merging

LLMs can produce inconsistent formulations of the same instrument name (e.g., "oh suppressing infra red imaging spectrograph" versus "oh suppressing infrared integral field spectrograph"). To enable matching of equivalent names, we normalized the LLM-provided canonical names using the following steps: **1.** Applied the same text normalization as in section 2.1.1; **2.** Expanded all unit abbreviations (e.g., "m" → "meter"); **3.** Removed parenthetical text.

We then semantically match potentially equivalent canonical names using a multi-step approach: **Token-sorted exact matching,** Split names on spaces, sort tokens alphabetically, and rejoin with spaces. Names with identical sorted token sequences were matched. This accounts for LLM variation in token ordering. **Fuzzy matching for long names,** For names with more than 4 tokens, we computed Jaro similarity scores and matched names with similarity > 0.95. We limited this to longer names to avoid incorrectly matching short names with different acronyms. **Quantity equivalence checking,** Any names containing numerical quantities were checked for physical equivalence (e.g., "400 cm" and "4 m" match, but "130 mm" and "2 m" do not). Matches with non-equivalent quantities were rejected.

For some candidates, the generated names differed not just in term ordering, but also in language or specificity. These types of naming variations are harder to resolve without a deeper understanding of the text. To remedy this, we conduct an extra semantic matching step consisting of: **Web search,** We search the open internet via the Brave Search API for each canonical name to ground further steps. **Instrument description generation,** For each name we generated a summary of the instrument in question using the web results from the previous step. These summaries included location, purpose, instrument type, and other information that might assist with disambiguation. **Description embedding matching,** The above instrument descriptions were embedded using a large text embedding model, then matched using a threshold of > 0.9 cosine similarity.

To verify the description-based matches, we used an LLM to make final matching decisions. The model was provided with the names and summaries of potentially

matching instruments and asked to return a list of lists, where each sublist contained instruments that genuinely refer to the same physical facility. This step helped reduce false positive matches, for example an array and a sub-component of the same array, from the automated similarity-based approach.

3. Results & Conclusions

The pipeline processed approximately 600,000 papers from the ADS/SciX corpus containing the term "telescope," extracting millions of n-gram candidates. Frequency filtering yielded approximately 11,000 unique n-grams (those beginning with "the" and appearing ≥20 times). After augmentation with web search results, LLM classification via majority voting across three independent queries identified approximately 1,700 n-grams as genuine instrument references.

Canonical name normalization consolidated variant names through a three-tiered matching strategy: token-sorted exact matching, fuzzy similarity for longer names, and quantity equivalence checking. Embedding-based description matching, followed by LLM verification, captured additional matches that were too dissimilar for string-based methods.

The complete pipeline runs in hours rather than the weeks required for manual curation, and is designed for incremental execution as new papers enter the corpus. By grounding LLM classification with web search, we achieved high-precision extraction while mitigating hallucinations.

Future work includes systematic curator assessment of results, automatic prompting methods, agent-based approaches, and incorporating additional context where licensing permits.

Acknowledgments. This research has made use of the Astrophysics Data System, funded by NASA under Cooperative Agreement 80NSSC21M00561.

References

Accomazzi, A., Henneken, E., Erdmann, C., & Rots, A. 2012, in Observatory Operations: Strategies, Processes, and Systems IV, edited by A. B. Peck, R. L. Seaman, & F. Comeron, vol. 8448 of Society of Photo-Optical Instrumentation Engineers (SPIE) Conference Series, 84480K. eprint: 1206.6352

Bartlett, J., Polimera, M., Lockhart, K., Accomazzi, A., Kurtz, M., & Science Explorer Team 2025, in American Astronomical Society Meeting Abstracts #245, vol. 245 of American Astronomical Society Meeting Abstracts, 442.04

Grant, C. S., Accomazzi, A., Eichhorn, G., Kurtz, M. J., & Murray, S. S. 2000, A&AS, 143, 111. eprint: astro-ph/0002103

Li, J., Gao, Y., Yang, Y., Bai, Y., Zhou, X., Li, Y., Sun, H., Liu, Y., Si, X., Ye, Y., et al. 2025, ACM Computing Surveys

Mayernik, M. S., Hart, D. L., Maull, K. E., & Weber, N. M. 2017, Journal of the Association for Information Science and Technology, 68, 1341

Observatory Bibliographers Collaboration, D'Abrusco, R., Gomez, M., Grothkopf, U., Hunt, S., Kneale, R., Konuma, M., Novacescu, J., Rebull, L., Scire, E., & et al. 2024, The Open Journal of Astrophysics, 7, 85. eprint: 2401.00060

Qwen Team 2025, Qwen3 technical report. eprint: 2505.09388, URL `https://arxiv.org/abs/2505.09388`

Tonmoy, S., Zaman, S., Jain, V., Rani, A., Rawte, V., Chadha, A., & Das, A. 2024, arXiv preprint arXiv:2401.01313, 6

Astronomical Data Analysis Software and Systems XXXV
*ASP Conference Series, Vol. **VOL** Compiled: Tuesday 30th June, 2026@ 01:30*
K. Polsterer, D. Bohlender, S. Gwyn, A. Hincks and P. Teuben, eds.
*© **YEAR** Astronomical Society of the Pacific*

ZZZDEBUG!243.tex
Oberto, A.|textbf
Landais, G

SIMBAD and Vizier: abuse of service crawling

Anaïs Oberto [1] and Gilles Landais [1]

[1]*Observatoire astronomique de Strasbourg, Université de Strasbourg, CNRS, UMR 7550, 67000 Strasbourg, France*

Abstract. The increasing integration of astronomical data services with machine learning workflows has led to unprecedented demand on web-based astronomical databases.

The SIMBAD astronomical database, operated by the Centre de Donnés astronomiques de Strasbourg (CDS), has experienced a significant surge in API requests, particularly from automated systems and AI model training pipelines that span multiple software ecosystems. We present an analysis of request patterns observed on the SIMBAD server, characterizing the emergence of untraceable automated queries that complicate load balancing and resource allocation.

Should we addresses strategies for sustainable collaboration between astronomical data providers and the broader computational science community, including rate limiting approaches, API design considerations, and infrastructure scaling solutions that accommodate diverse usage patterns across different software ecosystems and research disciplines.

1. Introduction

SIMBAD and Vizier astronomical databases are widely used by the research community since many years. These services are available online, without restriction or login. However, this public accessibility exposes the services to a heterogeneous mix of traffic: ordinary human queries, web-crawlers, AI-driven agents, and potentially abusive scrapers.

We will describe here the recent traffic trends, characterised by an increase in the number of distinct IP addresses, and no specific signature to identify the huge quantity of suspect queries (Casden et al. (2025)).

2. Usage statistics

The SIMBAD and Vizier astronomical databases, have experienced a significant surge in API requests, in number of queries, or IPs adresses. We are using here, the Apache HTTP access logs. Requests whose User-Agent strings explicitly mentioned known bots or crawler are were removed from the analysis.

2.1. Many queries

We observed a growth of number of queries reaching higher maxima per second (identified bots removed). The queries do not have a specific signature and are syntactically

correct queries. For instance, in SIMBAD, we went from 0.5M queries / month in February 2024 to 1.1M queries / month in Feburary 2025.

2.2. Many IPs origins

Monthly statistics (identified bots removed) reveal a huge number of different IP address accessing the service per month or per day. In mean, in SIMBAD, it has increase by a factor of 10 in few weeks (225K distinct IPs in February 2024, against 2.7M in July 2025). The increase is also big (x5) if we group them, disregarding the final IP number. This shows that this not only linked to subnets cloud IPs.

VizieR is experiencing similar behaviour, with a high number of IP addresses (+1489% in 2025).

2.3. Many countries

The distribution by country of IP origin of SIMBAD queries shows that the increase is strongly linked to the country but changes over time.

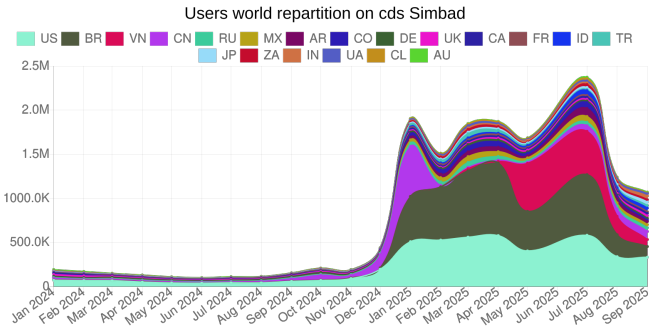


Figure 1. Countries identified from IPs adresses using SIMBAD service.

2.4. No user-agent

In this statistics, in SIMBAD logs, 95% of them do not have a web "referer", and the user-agent description is either empty or "Mozilla xxx-something-xxx". The quantity of queries having a user-agent identified as well known IA-bots "OpenAI / Anthropic / perplexity / FacebookBot / Mistral ..." (Tyler Einberger (2025)) are only about 6% of them.

3. Reactions

This is a real challenge to distinguish malicious scrapers from legitimate AI agents. We are currently starting a plan giving a protection of the real scientific usage of the services. SIMBAD and Vizier have to use strategies, including blacklists, HTTP tuning (apache) and application filtering based on request parameters.

We will investigate to add a login, or token, or captcha, at least in some cases, to slow down the quantity of bots.

4. Conclusion

In addition to traffic generated by bots or AI, the service may be subject to intensive activities from requests made by real users (e.g. >50 requests/second).

All the queries are syntactically correct but some bots do not have a distinctive signature, making selective filtering difficult without penalising legitimate users.

The type of filtering will be very important, and should be adaptative to the quantity of all the users.

References

Casden, J., Romani, D., Shearer, T., & Campbell, J. 2025, Code4Lib Journal. URL https://journal.code4lib.org/articles/18489/comment-page-1

Tyler Einberger 2025, AI user-agents, bots, and crawlers to watch. URL https://momenticmarketing.com/blog/ai-search-crawlers-bots

Astronomical Data Analysis Software and Systems XXXV
*ASP Conference Series, Vol. **VOL** Compiled: Tuesday 30th June, 2026@ 01:30*
K. Polsterer, D. Bohlender, S. Gwyn, A. Hincks and P. Teuben, eds.
*© **YEAR** Astronomical Society of the Pacific*

ZZZDEBUG!244.tex
Buchholz, E.|textbf
Heßling, H.
Haupt, L.

Memory-based Computing: First Results

Elsa Buchholz,[1] Lars Haupt,[1] and Hermann Heßling[1]
[1]*German Centre for Astrophysics (DZA), Görlitz, Saxony, Germany;*
elsa.buchholz@dzastro.de

Abstract. The Square Kilometre Array Observatory (SKAO) generates high-resolution 3D images of the universe in the petabyte range, which cannot be processed very efficiently with current processor-centric architectures due to the memory wall problem. This work investigates memory-based computing as an alternative architecture. The performance of a simplistic model for an I/O-dominated workflow is examined for network-mounted storage and compared with memory-based computing. The result is that memory-based computing speeds up the processing time by about two orders of magnitude. This result illustrates the potential of memory-based computing over conventional architectures, which are often used in astronomical workflows.

1. Introduction

The Square Kilometre Observatory (SKAO) is able to capture very high-resolution images of the universe, where individual images can be up to petabytes in size [1]. However, processors often stall because they are waiting for data, which can cause performance issues. This is known as the memory wall problem (Wulf & McKee 1995). Its significance increases when petabytes of data need to be processed. In order to address this issue with regard to the upcoming data from the SKAO, a paradigm shift is needed: Current processor-centric computing should be replaced by memory-based computing. The German Centre for Astrophysics (DZA) [2] is exploring this new type of computing architecture developed by Hewlett Packard Enterprise (HPE). This article presents performance results for a simplistic astronomical workflow analyzed on memory-based computing, using methods similar to those described in (Heßling et al. 2021). In astronomy, a common workflow is to read data from a hard disk during an imaging workflow. The performance of analyzing large amounts of data from hard disks mounted over Ethernet is compared with the performance of analyzing data using memory-based computing.

[1]SKA Observatory, Science Processing Centers, accessed 2025-11-20, URL: https://www.skao.int/en/explore/big-data/361/science-processing-centres.

[2]German Centre for Astrophysics, accessed 2025-11-23, URL: https://www.deutscheszentrumastrophysik.de/en.

2. Architecture and Methods

This work uses a prototype for a memory-based computing system called 'Julia'. 'Julia' is housed in a 19" rack and has a modular architecture: There are 8 chassis, each equipped with 4 CPUs, and each CPU has 28 cores and 1.5 TB of main memory (DRAM). The four CPUs within a chassis are connected to each other in ring via Ultra Path Interconnect (UPI) and are also directly connected to each CPU in the other chassis via HPE Superdome Flex Grid cables (Hewlett Packard Enterprise 2021). The bandwidths for random read operations of any CPU node from memory of all other CPU were determined using the Intel Memory Latency Checker (Corporation 2025). As can be extracted from Fig. 1 (middle), reading from main memory directly attached to a CPU is fastest at 93 GB/s, while reading from the main memory of other CPUs in the same chassis is significantly slower at 17 GB/s. Reading from the main memories in a remote chassis via Flex Grid is the slowest, but at 12 GB/s it is significantly faster than accessing a remote hard disk storage via a 10 Gbps network connection.

For the analyses described below, two chassis with a configuration as shown in Fig. 1 (left), were used. The CPUs in one chassis were not used, only their main memories ('fabric') were used to create two shared memories, see Fig. 1 (left bottom). Each shared memory has 3 TB of storage space, which the CPUs in the other chassis ('compute') can access via a file mount and at 12 GB/s.

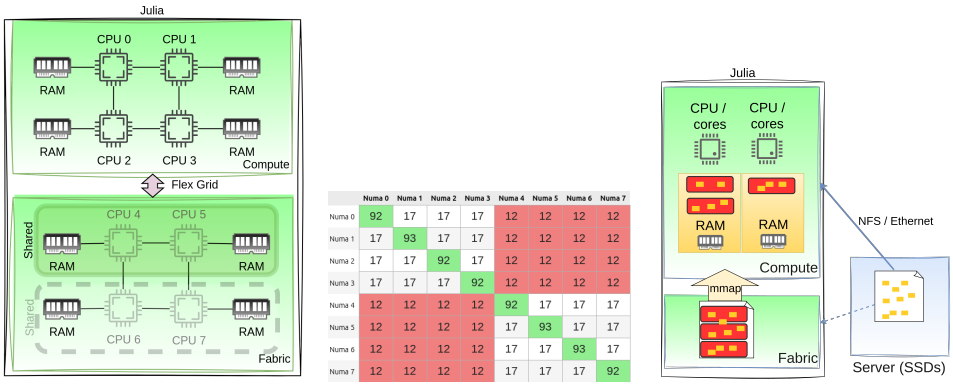


Figure 1. Left: The CPUs on the top are used for compute and the main memories of the CPUs at the bottom are combined to two shared memories ('fabric'). Middle: Performance of connections between CPUs in the memory-based system: The average bandwidth [GB/s] for random read operations from each NUMA node to all other NUMA nodes are determined using Intel's Memory Latency Checker. Right: Workflow based on a simplistic model of an SQL-based analysis. The data is read in small chunks [size 1KB] that are randomly distributed in a file, filled with random digits [see yellow boxes], and - to imitate an analysis - the sum of the digits in the chunks is calculated. First, according to the traditional workflow pattern, the file stored on a remote server is accessed via NFS (over Ethernet). Alternatively, the file is copied to the 'fabric' of Julia, and large subsets of the file ('windows' see red boxes) are provided from there for in-memory computation (via `mmap`), where the 'windows' are analyzed in parallel by threads. Illustration by the authors.

A simplistic SQL-like workflow is used to compare the performance between a traditional file read and a memory-mapped file read; see Fig. 1 (right). To ensure that reads are performed directly from the physical disk and fabric, without the effects of

cached data affecting performance, the page cache of the operating system (OS) is dropped. The `fread` system call is commonly used for reading files from disk. This benchmark operates in a single-thread mode and reads each chunk sequentially. The second benchmark, which involves reading a memory-mapped file, uses the `mmap` system call. In addition, the file is divided into 'windows' for `mmap`, which are locked into memory using the `mlock2` method. This ensures that the mapping stays in DRAM and cannot be read from the underlying 'fabric' (Kerrisk 2025).

3. Results and Discussion

In astronomy, data are often read from a remote disk storage that is mounted over a network. Fig. 2 (left) shows the results of analyzing the simplistic workflow for a file accessed via Network File System (NFS) over Ethernet. Only one thread was used for the analysis, but with repetitions at different times. The error bars show the standard deviations around the mean values. The strong fluctuations of the mean values could be caused by temporary high network utilization.

Fig. 2 (right) shows the results of the corresponding analyses of the same simplistic workflow on memory-based computing. First of all, it should be noted that the times are reduced by about two orders of magnitude, i.e. I/O-dominated workflows can benefit significantly from memory-based computing. Furthermore, the relatively small error bars show that no temporary strong fluctuations are to be expected in 'Julia's' closed system. In addition, the number of threads was varied, resulting in a further increase in performance. It is important to note in this context that the speedup is less than linear when the number of threads is increased. However there is no noticeable drop in performance with more than 8 threads, as was observed, for example when running the astronomy tool CASA (Common Astronomy Software Applications) in parallel (Buchholz 2020).

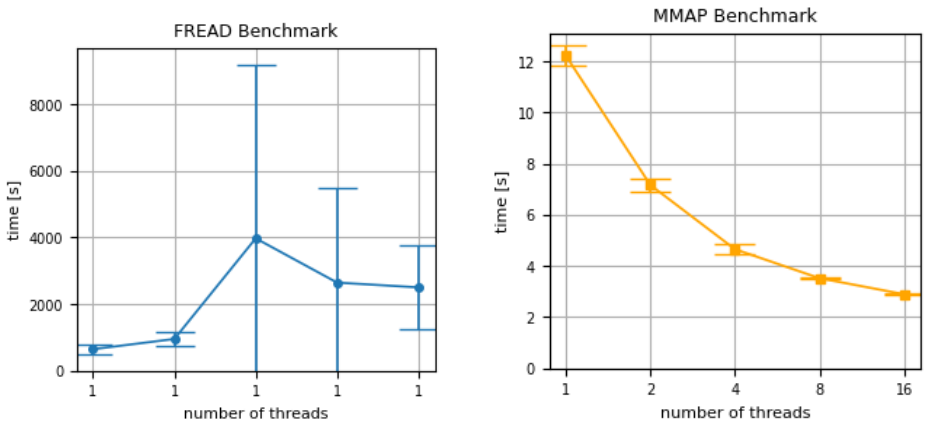


Figure 2. Benchmark results of the simplistic model: The time for analysing a 1.5 TB file with 1 millionen chunks (y-achsis) versus the number of threads (x-achsis), the error bars show the standard-deviation. Left: Conventional analysis where the file is stored on a mounted remote disk space. Right: Memory-based analysis where the file is stored in a fabric. Memory-based computing leads to a speedup of the time by nearly two orders of magnitude.

To summarize, a comparison between a traditional file read and memory-mapped file read shows that the latter achieves a speedup of nearly two orders of magnitude. Memory-based computing enables faster access to large files than the traditional `fread` method. This performance improvement is relevant for the efficient processing of petabyte-scale data volumes produced by the SKAO. The impact of memory-based computing on an astronomical workflow will be explored next. Large files must be divided into smaller 'windows' so that they fit into the main memory of individual CPUs in the 'compute' area, which could lead to data loss. The impact of this partitioning on multithreaded analysis will be examined next, as well as the impact of using Zarr files (Miles et al. 2023).

Acknowledgments. We are grateful for stimulating discussions with Sagar Borra (DZA Görlitz), Marc Drobek (DZA Görlitz), Andrei Kazantsev (MPIfR Bonn), Yurii Pidopryhora (MPIfR Bonn), Tanumoy Saha (HTW Berlin), Marcel Trattner (HTW Berlin), Yanik Yiannakis (MPIfR Bonn). We would like to thank the Center for Information Services and High Performance Computing (ZIH) of TU Dresden for providing computational resources and support. We would also like to express our sincere gratitude to Clarete Riana Crasta from HPE for her invaluable assistance in enhancing our understanding of the memory-based computing architecture.

References

Buchholz, E. 2020, Master's thesis, University of Applied Sciences (HTW) Berlin, Berlin. Submitted in fulfillment of the requirements for the degree of Master of Science (M.Sc.) in Applied Computer Science, Department of Computer Science, Communication, and Business

Corporation, I. 2025, Intel Memory Latency Checker (Intel MLC), Intel Corporation. Accessed: 2025-11-23, URL www.intel.com/content/www/us/en/download/736633/intel-memory-latency-checker-intel-mlc.html

Heßling, H., Strutz, M., Buchholz, E. I., & Hufnagl, P. 2021, Big Data Res., 25. URL https://doi.org/10.1016/j.bdr.2021.100214

Hewlett Packard Enterprise 2021, HPE Superdome Flex Server Architecture and RAS, Tech. rep., Hewlett Packard Enterprise. Accessed: 2025-11-23, URL https://hpe.metroconnect.co.th/wp-content/uploads/2023/11/HPE-Superdome-Flex-Server-Architecture-and-RAS-technical-white-paper-a00036491enw.pdf

Kerrisk, M. 2025, mlock2 Linux manual page, man7.org Training and Consulting. Accessed: 2025-11-23, URL www.man7.org/linux/man-pages/man2/mlock.2.html

Miles, A., Striebel, J., Rzepka, N., Maitin-Shepard, J., & Moore, J. 2023, Zarr Core Specification, Version 3.1. Accessed: 2025-12-13, URL https://zarr-specs.readthedocs.io/en/latest/v3/core/index.html

Wulf, W. A., & McKee, S. A. 1995, ACM SIGARCH computer architecture news, 23, 20

Astronomical Data Analysis Software and Systems XXXV
*ASP Conference Series, Vol. **VOL** Compiled: Tuesday 30th June, 2026@ 01:30*
K. Polsterer, D. Bohlender, S. Gwyn, A. Hincks and P. Teuben, eds.
*© **YEAR** Astronomical Society of the Pacific*

ZZZDEBUG!245.tex
Barloy, N.|textbf
Landais, G.
libraries!Astropy
data formats!FITS
protocols!ObsCore
Saada! ascl:1111.003
Astropy! ascl:1304.002

Creating an ObsCore builder for data submission

Nathan Barloy and Gilles Landais

CDS, Strasbourg, Grand-Est, France

Abstract. The catalogs in the CDS VizieR service (Ochsenbein et al. 2000) need to have an ObsCore table associated to them to be easily referenced. But the quantity of new catalogs we receive is huge compared to the number of staff we have on it. That is why we are creating a tool to automatically generate those ObsCore tables from the FITS files of the catalogs. This tool is for now a Python library that will extract a maximum of information from the headers of the FITS files. Once this is done, it is possible to curate the information with external services. Furthermore, since a non-negligible number a FITS files don't follow the standard but make their own, we added a way to custom the way information will be extracted with a specification file we call a Mapping file. For now this tool is aimed to help the ingestion process in the CDS VizieR service, but if more persons are interested, we could try to make an open version of this.

1. Context

1.1. Reason

The number of catalogs we keep receiving for VizieR at the CDS is always growing. A lot of them are not integrated in our ObsCore database (called assocdata (Ocvirk et al. 2018)), so it can be difficult to access them from searches. We have the ODAS project (Open Data for AStronomy) at CDS, and we want to have more catalogs indexed in the ObsCore database. The problem is, the process is not easy, and our staff is already overworked, there is always more catalogs that are submitted to VizieR. We want to try to automate this process at best, to unburden the load of work of our staff The scope could be expanded if it works well : trying to have an open version of this, that other people will be able to use. But for now, we are really focused to have a tool to help our process.

1.2. The ODAS project

The CDS-ODAS project (Open Data for AStronomy) is a 3-year project with the purpose of making the VizieR data more FAIR (Findable, Accessible, Interoperable and Reusable). We already follow the IVOA standards, but not enough of our data is easily accessible via some ObsCore information. The main focus will logically be to have a tool allowing for more data to be added in our ObsCore database, and to create a better workflow for data submission.

1.3. Purpose

The ObsCore Builder library is a Python library that gives tools to create an ObsCore table from a collection of FITS files. The library aims to replace the current worklows which is based on Saada software (Landais et al. 2019). The Saada software enabled us to build a spectrum and image database available in an ObsCore table (> 1M spectra and 50K images) However, in order to anticipate the long-term maintenance of our service and to better integrate the service int our architecture, we decided to develop a new API in Python. This library can not only extract information from the FITS files, but it can also be given complementary information from the user in the form of a Mapping file. Using this file, the library can also apply those complementary information to an already existing ObsCore table. Furthermore, a Python library is very versatile and could be used in a lot of cases. For example, making a CLI would not be very difficult, and creating a back-end service would be very useful to create a new submission flow for new catalogs.

2. How does it work ?

2.1. Requirements

This Python library uses Python 3.11. It is heavily based on the AstroPy library (Astropy Collaboration et al. 2013, 2018, 2022).

2.2. Extraction

The ObsCore extraction from a FITS collection is the main feature of this library. Since the catalogs we receive can be very large, we have to be as efficient as possible when extracting the information. That is why only the headers of the files will be opened. The extraction process is composed of 3 main steps :

1. Extract information from the FITS keywords. For some ObsCore fields, we just use a list of common FITS keywords that can contain the useful information. Those lists of common FITS keywords are in a configuration file, and can be modified/improved if needed.

2. Solve the WCS. Those are often the most important information of the ObsCore (s_ra and s_dec for an image, em_min and em_max for a spectrum).

3. Clean the results. We have to make sure that each value we got can be converted in the right format for ObsCore.

Note: There is often a dilemma about how much non-standard information we should accept and understand. The general solution we use is that we can support non-standard keywords and notations as long as it doesn't conflict with the FITS standards.

2.3. Mapping file

Sometimes, we know that the FITS files will follow their own rules, and have their own keywords. Sometimes, there are information missing in the FITS keywords, like a telescope or a wavelength, but we know what it should be from other sources. That is why it is useful to have a way for the user to indicate more specific information:

- Default values
- Missing FITS keywords
- Define your own functions
- ...

Our Mapping file format is a way to achieve that.

The format is a simple list of *key: value* pairs. The *key* can be either an ObsCore field or a FITS keyword. The *value* can be one of the following things:

- A simple value (integer, float, string)
- A FITS keyword
- An other ObsCore field
- An environment variable (in a list of predefined special values, like the file name)
- A function (either a predefined function available, or a custom one)
- A list of values

When using a Mapping file in an extraction, the given FITS keys will be replaced by their associated values in the header, and the ObsCore fields will be replaced by their associated values in the resulting ObsCore table.

There are two ways of associating a key with a value: a *forced* value and a *default* value. With a forced value, the value given in the Mapping file will override whatever value we could have had with the extraction. With a default value, the value given in the mapping file will be used only if no other value has been found in the extraction.

All of this allows for a precise way to generate an ObsCore table as wanted.

2.4. Usage

There is two functions in this library. The first one extracts an ObsCore table from a FITS collection and an optional Mapping file. The second one applies a Mapping file to an already existing ObsCore table to modify it. This is illustrated in the Figure 1.

For this second function, we don't have the original FITS files anymore, so we can't use any reference to the FITS keywords or the environment variables. If those values are used in a line of the Mapping file, this line will simply be ignored.

References

Astropy Collaboration, et al. 2013, A&A, 558, A33. eprint: 1307.6212
— 2018, AJ, 156, 123. eprint: 1801.02634
— 2022, ApJ, 935, 167. eprint: 2206.14220
Landais, G., Michel, L., Derriere, S., Ocvirk, P., & Royer, F. 2019, in Astronomical Data Analysis Software and Systems XXVI, edited by M. Molinaro, K. Shortridge, & F. Pasian, vol. 521 of Astronomical Society of the Pacific Conference Series, 458
Ochsenbein, F., Bauer, P., & Marcout, J. 2000, A&AS, 143, 23. eprint: astro-ph/0002122
Ocvirk, P., Landais, G., Michel, L., Arab, H., Guehenneux, S., Boch, T., Brouty, M., Perret, E., Pineau, F.-X., Pouvreau, T., & Vannier, P. 2018, in European Physical Journal Web of Conferences (EDP), vol. 186 of European Physical Journal Web of Conferences, 02002

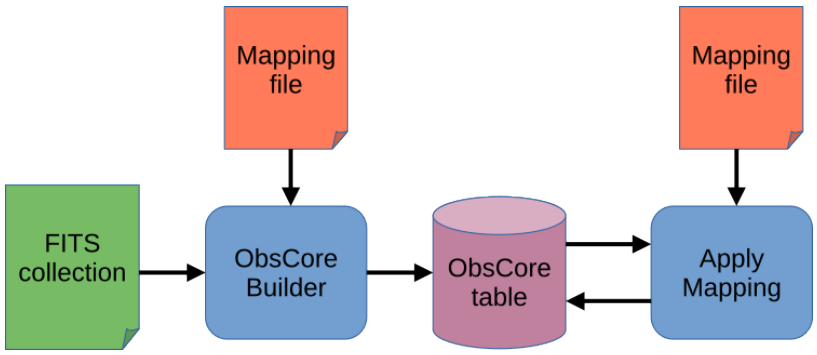


Figure 1. The usage pipeline.

Astronomical Data Analysis Software and Systems XXXV
*ASP Conference Series, Vol. **VOL** Compiled: Tuesday 30th June, 2026@ 01:30*
K. Polsterer, D. Bohlender, S. Gwyn, A. Hincks and P. Teuben, eds.
*© **YEAR** Astronomical Society of the Pacific*

ZZZDEBUG!246.tex
Bourgès, L.|textbf
Mella, G.
Chelli, A.
Duvert, G.
Cruzalèbes, P.
Berger, J.-P.
Tallon-Bosc, I.
Mourard, D.
FOOBAR!conference!ADASS 2019
FOOBAR!organisations!ASP
FOOBAR! ascl:1101.010

The JMMC Stellar Diameter Catalog version 3, a complete open-source, reproducible & open-data analysis pipeline

L. Bourgès,[1] G. Mella,[1] A. Chelli,[2] G. Duvert,[3] P. Cruzalèbes,[2] J-P. Berger,[3] I. Tallon-Bosc,[4] and D. Mourard[2]

[1]*Univ. Grenoble Alpes, CNRS, IRD, INRAE, Météo France, OSUG, Grenoble, 38000, France; laurent.bourges@univ-grenoble-alpes.fr*

[2]*Université Côte d'Azur, Observatoire de la Côte d'Azur, CNRS, Laboratoire Lagrange, Nice, 06304, France*

[3]*Univ. Grenoble Alpes, CNRS, IPAG, Grenoble, 38000, France*

[4]*Univ Lyon, Univ Lyon1, Ens de Lyon, CNRS, CRAL, Saint-Genis-Laval, 69230, France*

Abstract. The Jean-Marie Marrioti Center (JMMC) is providing the JMMC Stellar Diameter Catalog (JSDC) releases integrated in SearchCal & GetStar web services. For the JSDC 3 release, the software package is available as an open-source repository able to process the JMDC catalog and fit stellar pseudo-magnitude brightness relations, collect and cross match all available information on 2.5 million stars in GAIA DR3, 2MASS, All-Wise... catalogs from Vizier & SIMBAD archives, estimate stellar diameters to produce the JSDC 3 dataset, the new reference to improve the theoretical approach to build the JSDC 4 to be compared with the GAIA DR4.

1. JMMC mission & services

Since 2000, the Jean-Marie Marrioti Center[1] (JMMC) aims to coordinate the efforts of French observatories involved in Optical Interferometry (OI) to offer the best operational environment (within the Virtual Observatory) to all potential users of OI facilities, i.e. the VLTI[2]/ESO and CHARA[3]/GSU instruments. Its mission consists in 1) developing, producing, documenting and maintaining software & web services necessary for the exploitation of the current instruments, 2) providing a "Face to Face" User support from the observation preparation, analysis to the data archiving, 3) participating actively to the academic formation (VLTI school, video courses and tutorials), 4) providing and maintaining an interactive interferometry publication database (OLBIN) and 5) participating to the prospective around new interferometric instruments...

[1]The JMMC website is available at https://www.jmmc.fr/. Please contact the user support (and authors) at jmmc-user-support@jmmc.fr for any question.
[2]VLTI: Very Large Telescope Interferometer, see https://www.eso.org/sci/facilities/paranal/telescopes/vlti.html
[3]CHARA: Center for High Angular Resolution Astronomy, see https://www.chara.gsu.edu/

The JMMC Calibrator group develops methods to ascertain the angular diameter of stars since 25 years and provides this expertise in the SearchCal software and associated catalogs. The JMMC maintains on a regularly basis the JMMC Measured Diameter Catalog[4] (JMDC) and provides the JMMC Stellar Diameter Catalog[5] (JSDC), and an open-access to the JMMC SearchCal[6] graphical interface & the GetStar web service to respectively query & filter calibrator objects and get all information on a single object. The first JSDC catalog published in 2010 has ~38,000 stars, the JSDC 2 (Bourgès et al. 2014) published in 2017 has ~465,000 stars, based on the robust approach published in (Chelli et al. 2016).

2. The updated pipeline for the JSDC 3 catalog

For the JSDC 3 release, the complete software package is available since 2025 as an open-source repository[7] to perform the complete & reproducible data analysis workflow: 1) Use the JMMC GetStar service to complete the JMDC dataset, filter non-standard objects and fit the stellar pseudo-magnitude brightness relations (V-J, V-H, V-K colors) (GDL[8] script), 2) Prepare JSDC catalog candidates by cross-matching the TYCHO-2 (ASCC) with the CDS SIMBAD database (Spectral and Object types) and the MDFC catalog (Cruzalèbes et al. 2019) (CDS xmatch + STILTS), 3) Run the SearchCal server (C/C++) to execute either the JSDC BRIGHT or FAINT scenario on candidates (respectively 525k with known spectral type and 1.95m stars without), 4) Filter the output to produce the final table (proper metadata).

Each SearchCal scenario performs the following steps: 1) Collect and cross match all available information on JSDC candidates from TYCHO-2, GAIA DR3, 2MASS, All-Wise... catalogs from the CDS Vizier service, 2) Estimate theoretical stellar diameters using the pseudo-magnitude brightness relations using VJHK fluxes. 3) Export the output catalog (100 properties) in the VOTable format.

Both SearchCal & JSDC 3 use the same polynomial coefficients for pseudo-magnitude brightness relations as the JSDC 2. The faint approach uses a brute-force estimation of stellar diameters without the spectral type (sampling all range) with higher uncertainty. Finally the SearchCal & GetStar server (C/C++) are built using a docker recipe (ubuntu 24.04) and the JMMC infrastructure uses a Kubernetes cluster (only 8 gb RAM per pod) to run services (NFS storage for the JSDC data files ~15 gb

The SearchCal 6 GUI & GetStar 6 service use the new SearchCal Server including lots of enhancements (improved crossmatch algorithm with neighborhood counts, updated calibrator flags, GAIA fluxes) and the JSDC 3 dataset (stored as a local cache).

3. Updated JSDC scenario

The JSDC3 scenario uses SIMBAD (2025.01 ~ 20e6 objects), GAIA DR3 and MDFC catalogs to refine star coordinates, get GAIA fluxes, distance & radius and retrieve the

[4]The JMDC catalog (CDS Vizier Cat. II/345) is available at https://jmdc.jmmc.fr/

[5]The JSDC catalog (V2 at CDS Vizier Cat. II/346) is available at https://jmmc.fr/jsdc (more files)

[6]JMMC software and services are listed on the release page at https://releases.jmmc.fr/

[7]See the SearchCal repository at https://github.com/JMMC-OpenDev/SearchCal. All JMMC open-source repositories (GPL2) are available at https://github.com/JMMC-OpenDev/.

[8]GDL: GNU Data Language, see: https://gnudatalanguage.github.io/

mid-infrared flag (MDFC): see figure 1. Each JSDC star properties preserve the catalog origin and has a confidence index (low to high).

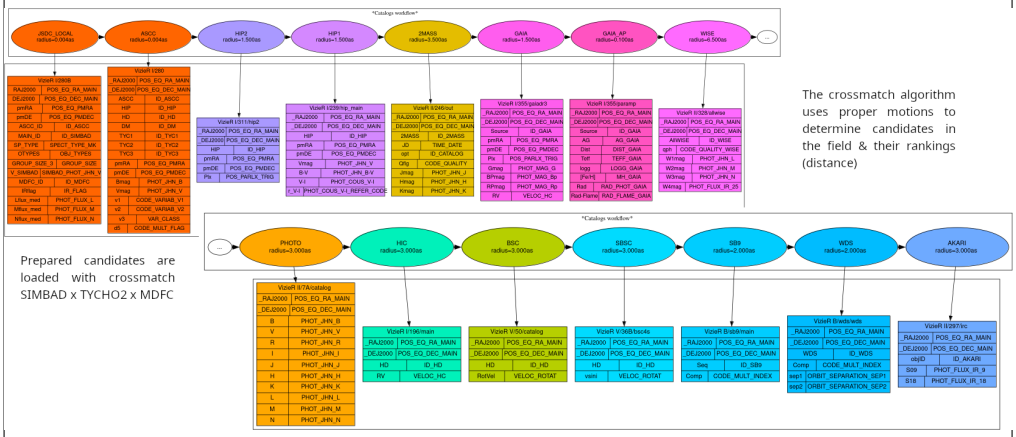


Figure 1. The JSDC 3 pipeline showing queried catalogs with their columns.

4. JSDC 3 statistics

The JSDC 3 contains 514k diameters vs 465k (JSDC 2) where diameters are identical (same JMDC polynomials), except for few thousands where the cross-identification or the spectral type (SIMBAD) changed. The figure 2 shows flux histograms, relative error (min 2%) on JSDC diameters and the source counts per spectral type.

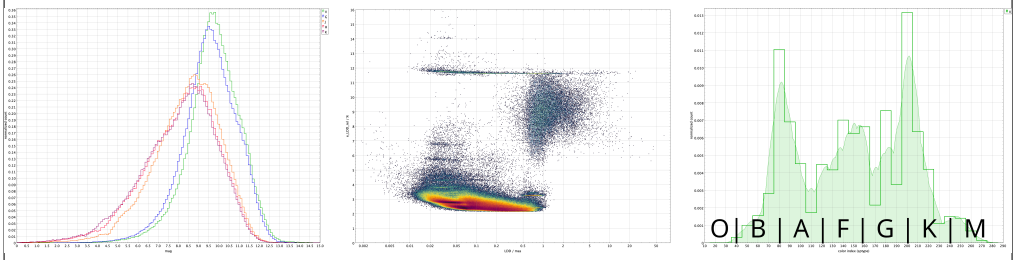


Figure 2. *Left:* Histogram of [V G J H K] fluxes. *Middle:* Relative error (%) vs LDD in log scale. *Right:* Histogram of color index (Spectral type)

5. JSDC 3 vs GAIA DR3 diameter comparison

There is a very good agreement between JSDC3 and GAIA diameters (converted from GAIA radius and distance) within 1σ on 324k cross-matched stars: see figure 3.

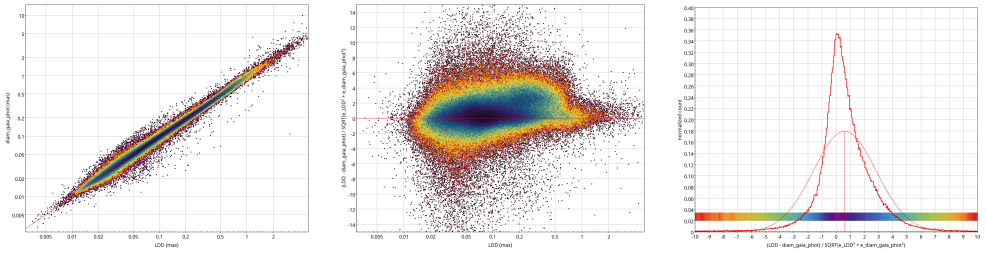


Figure 3. *Left:* GAIA diameter (phot) vs JSDC3 LDD (mas) in log scale. *Middle:* Residuals between GAIA vs JSDC3 diameter (σ). *Right:* Histogram of residuals (σ)

6. Perspectives

The JMMC technical team is working with the French SPICA/CHARA team for an efficient and programmatic survey scheduling (ERC ISSP[9]). The SPICA instrument will provide valuable diameters for more precise future JSDC releases!

In short terms, the JSDC team will compare cross-identification of sources between the JSDC, SIMBAD (identifiers) and GAIA crossmatches, improve the Faint approach to use the GAIA distance & effective temperature to constrain the spectral type, hence the stellar diameter possible range.

Finally enhancing the pseudo-magnitude brightness relations (filtering correlations in JMDC and using more colors with Gaia and mid-IR fluxes) is the new priority for the JSDC 4 release dedicated to compare with the GAIA DR4 data release.

Acknowledgments. This research has made use of the SIMBAD database and VizieR catalogue access tool, CDS (Strasbourg, France) and astronomical software tools TOPCAT and (Taylor 2006, STILTS) too.

References

Bourgès, L., Lafrasse, S., Mella, G., Chesneau, O., Le Bouquin, J.-B., Duvert, G., Chelli, A., & Delfosse, X. 2014, in Astronomical Data Analysis Software and Systems XXIII, edited by N. Manset, & P. Forshay, vol. 485 of Astronomical Society of the Pacific Conference Series, 223

Chelli, A., Duvert, G., Bourgès, L., Mella, G., Lafrasse, S., Bonneau, D., & Chesneau, O. 2016, A&A, 589, A112. eprint: 1604.07700

Cruzalèbes, P., Petrov, R. G., Robbe-Dubois, S., Varga, J., Burtscher, L., Allouche, F., Berio, P., Hofmann, K.-H., Hron, J., Jaffe, W., Lagarde, S., Lopez, B., Matter, A., Meilland, A., Meisenheimer, K., Millour, F., & Schertl, D. 2019, MNRAS, 490, 3158. eprint: 1910.00542

Taylor, M. B. 2006, in Astronomical Data Analysis Software and Systems XV, edited by C. Gabriel, C. Arviset, D. Ponz, & S. Enrique, vol. 351 of Astronomical Society of the Pacific Conference Series, 666

[9]ISSP: Interferometric Survey of Stellar Parameters, see https://lagrange.oca.eu/fr/issp-welcome.

Astronomical Data Analysis Software and Systems XXXV
*ASP Conference Series, Vol. **VOL** Compiled: Tuesday 30th June, 2026@ 01:30*
K. Polsterer, D. Bohlender, S. Gwyn, A. Hincks and P. Teuben, eds.
*© **YEAR** Astronomical Society of the Pacific*

ZZZDEBUG!247.tex
Kretlow, M.|textbf
computer languages!Julia
FOOBAR! ascl:1101.010

The Julia Ecosystem for Astronomy, Astrophysics, and Geospatial Data Science in 2025

Mike Kretlow[1]

[1]*Deutsches Zentrum für Astrophysik (DZA), 02826 Görlitz, Germany;*
mike.kretlow@dzastro.de

Abstract. The Julia programming language has matured into a powerful platform for scientific computing, combining high-level expressiveness with performance comparable to traditionally compiled languages. Over the last decade, its ecosystem has expanded rapidly, including broad support for astronomy, astrophysics, and geospatial data science. This paper provides a concise overview of the Julia ecosystem as of 2025, focusing on packages relevant to astronomical data analysis, modeling, and large-scale computation. Rather than aiming for completeness, the paper highlights representative tools and experiences from an active user's perspective.

1. Introduction

Modern astronomy increasingly relies on complex numerical modeling, large-scale data processing, and interactive exploratory analysis. Traditionally, these tasks have been split across multiple programming environments: high-level languages such as Python or MATLAB for prototyping and visualization, and low-level languages such as C, C++, or Fortran for performance-critical components. While effective, this division often leads to fragmented workflows, duplicated implementations, and significant maintenance overhead.

Julia was designed explicitly to address this "two-language problem" by offering a high-level, dynamically typed language with execution speeds approaching those of compiled languages (Bezanson et al. 2017). Since its public release in 2012, Julia has gained steady adoption in scientific computing, and by 2025 it has developed a rich ecosystem covering much of the typical astronomical software stack.

This contribution summarizes the current state of the Julia ecosystem for astronomy, astrophysics, and geospatial data science, illustrating how Julia can serve as a unified environment for research-grade analysis, algorithm development, and production workflows.

The rapid expansion of Julia's package ecosystem over the last five years is illustrated in Fig. 1. The Julia general registry grew from roughly 3,000 packages in 2020 to more than 11,000 packages by 2025, indicating sustained community activity and increasing coverage across scientific and technical domains. Although raw package counts alone do not measure software quality or long-term maintenance, the consistent growth reflects a healthy and expanding ecosystem that provides a strong foundation for domain-specific development in astronomy, astrophysics, and geospatial data science.

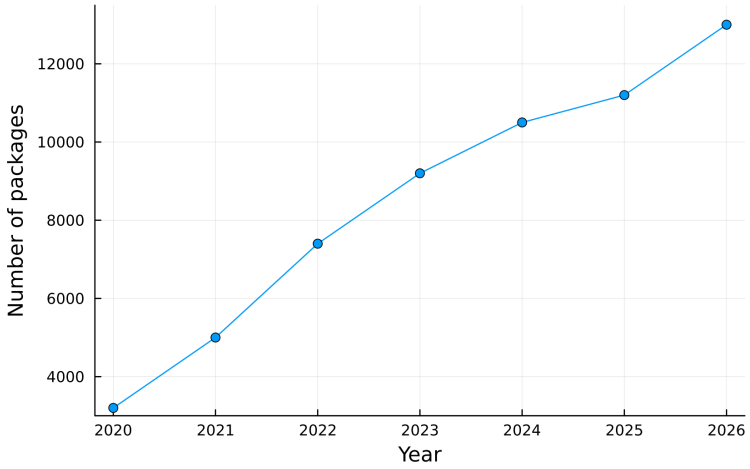


Figure 1. Growth of the Julia general registry between 2020 and 2025, shown as the total number of registered packages per year. While package count alone is not a direct proxy for scientific maturity or code quality, the sustained growth reflects increasing community adoption and ecosystem breadth.

2. Julia as a Scientific Programming Language

Julia combines an expressive, MATLAB-like syntax with a modern compiler infrastructure based on LLVM. Performance is achieved through just-in-time (JIT) compilation and aggressive specialization via multiple dispatch, allowing generic code to be compiled into highly optimized machine code for specific argument types. Key language features include: (1) multiple dispatch, allowing function behavior to be specialized on all argument types and enabling clean, extensible APIs for physical models and numerical methods; (2) native support for parallelism, including multithreading and distributed computing; (3) seamless foreign-function interfaces, allowing direct calls to C, C++, and Fortran libraries without glue code; (4) a built-in package and environment system (Pkg.jl), supporting reproducible science workflows.

Interactive computing is a major strength of Julia. In addition to a powerful REPL, Julia integrates smoothly with Jupyter notebooks, Pluto.jl for reactive notebooks, and web-based presentation tools such as BonitoBook.jl. This makes Julia equally suitable for exploratory analysis, teaching, and large-scale pipeline development. Plotting and visualization are enabled by Plots.jl and Makie.jl with publication-ready output.

Recent developments such as JuliaC.jl, introduced with Julia 1.12, enable Julia code to be compiled into self-contained native executables and shared libraries. This significantly lowers deployment overhead and addresses long-standing demands for easily distributable, stand-alone scientific applications.

3. Core Ecosystem and General-Purpose Packages

The Julia ecosystem provides a broad set of general-purpose scientific packages that form the foundation for astronomy and geospatial applications. These include libraries for numerical integration, optimization, probabilistic modeling, automatic differentia-

Table 1. Selection of Julia packages relevant for astronomy, astrophysics, and geospatial data science (2025). The list is representative rather than exhaustive and emphasizes widely used or foundational components.

Domain	Category	Representative Packages
General Scientific Computing	Mathematics, optimization, inference	DifferentialEquations.jl, Optim.jl, Optimization.jl, Distributions.jl, Turing.jl, Symbolics.jl, ForwardDiff.jl, Zygote.jl, Enzyme.jl
	Data handling and I/O	DataFrames.jl, Tables.jl, CSV.jl, HDF5.jl, JLD2.jl, Parquet.jl, Zarr.jl, FileIO.jl
	HPC and acceleration	CUDA.jl, AMDGPU.jl, Dagger.jl, Distributed.jl
Astronomy and Astrophysics	Constants, units, time	PhysicalConstants.jl, Unitful.jl, UnitfulAstro.jl, AstroTime.jl
	Coordinates and astrometry	Astrometry.jl, ERFA.jl, EarthOrientation.jl, WCS.jl, AstroAngles.jl, SuperNOVAS.jl
	Ephemerides and dynamics	SPICE.jl, CALCEPH.jl, HORIZONS.jl, GeneralAstrodynamics.jl
	Imaging and photometry	FITSIO.jl, CFITSIO.jl, AstroImages.jl, CCDReduction.jl, Photometry.jl
	Spectroscopy and time series	Spectra.jl, SpectralFitting.jl, LombScargle.jl, TimeSeries.jl
Geospatial Data Science	Raster and vector data	GDAL.jl, ArchGDAL.jl, GeoJSON.jl, Shapefile.jl, GeoDataFrames.jl
	Gridded and netCDF data	NetCDF.jl, NCDatasets.jl
	Coordinate systems and mapping	Proj.jl, Geodesy.jl, GeoMakie.jl, GMT.jl

tion, and symbolic computation, as well as mature solutions for tabular data handling and scientific file formats. Representative packages are summarized in Table 1.

4. Astronomy and Astrophysics Packages

Julia's astronomy and astrophysics ecosystem covers many standard tasks encountered in observational and theoretical research, including time systems, coordinate transformations, ephemerides, astrometry, image processing, photometry, and time-series analysis (Table 1). Many packages provide Julian interfaces to well-established external C or Fortran libraries, such as ERFA, SPICE, or CFITSIO, allowing users to combine legacy standards with modern, high-level workflows.

5. Geospatial Data Science

Geospatial data science is increasingly relevant for astronomical applications, particularly in observatory operations, survey planning, and Earth-based observation geometry. Julia's geospatial ecosystem provides tools for raster and vector data handling, coordinate reference systems, gridded data formats, and high-quality cartographic visualization. The close integration of geospatial packages with Julia's numerical and visualization stack allows seamless combination of geographic and astronomical computations within a single environment. Representative geospatial packages relevant to scientific workflows are included in Table 1.

6. Scientific Machine Learning

Scientific Machine Learning (SciML) combines data-driven approaches with physically motivated models and has become an important paradigm across computational science. By 2025, the Julia ecosystem provides mature packages for general-purpose neural networks (e.g. Lux.jl, Flux.jl), Physics-Informed Neural Networks (PINNs; e.g. NeuralPDE.jl, DiffEqFlux.jl), as well as optimization and probabilistic inference (e.g. Turing.jl). While these tools are also highly relevant for astrophysical applications, SciML constitutes a general methodology used across many scientific domains and is therefore not included in the package overview in Table 1.

7. Discussion and Outlook

While Julia is not yet as widely adopted in astronomy as Python, its ecosystem has matured to the point where it can support end-to-end scientific workflows. Remaining challenges include broader institutional adoption as well as the continued development, maintenance, and stabilization of packages and APIs; however, Julia's ability to unify prototyping, high-performance numerics, and interactive analysis makes it well suited for data-intensive astronomical projects as survey sizes and algorithmic complexity continue to grow.

Independent evaluations in high-energy physics show that Julia meets nearly all functional and performance requirements for event-based workloads, matching or exceeding C++ performance while significantly outperforming Python in throughput-critical sections (Eschle et al. 2023). Its suitability for large-scale scientific computing is further demonstrated by the Celeste.jl project, which processed approximately 178 TB of astronomical imaging data on 8192 nodes and achieved sustained performance of 1.54 PFLOP/s during Bayesian inference (Regier et al. 2019).

8. Conclusions

By 2025, Julia has established itself as a serious contender for astronomy, astrophysics, and geospatial data science. Its combination of performance, expressiveness, and interactive tooling allows researchers to develop complex numerical methods and data-analysis pipelines within a single coherent language. The growing ecosystem of astronomy, astrophysics and geospatial packages demonstrates that Julia is no longer experimental in this domain, but a viable and productive platform for both individual researchers and large collaborations.

References

Bezanson, J., Edelman, A., Karpinski, S., & Shah, V. B. 2017, SIAM Review, 59, 65

Eschle, J., Gál, T., Giordano, M., Gras, P., Hegner, B., et al. 2023, Computing and Software for Big Science, 7

Regier, J., Fischer, K., Pamnany, K., Noack, A., Revels, J., Lam, M., Howard, S., Giordano, R., Schlegel, D., McAuliffe, J., Thomas, R., & Prabhat 2019, Journal of Parallel and Distributed Computing, 127, 89

Astronomical Data Analysis Software and Systems XXXV
*ASP Conference Series, Vol. **VOL** Compiled: Tuesday 30th June, 2026@ 01:30*
K. Polsterer, D. Bohlender, S. Gwyn, A. Hincks and P. Teuben, eds.
*© **YEAR** Astronomical Society of the Pacific*

ZZZDEBUG!248.tex
Greiner, B.|textbf
Bryant, A.
Früh, J.
Hütwohl, M.
Iserlohe, I.
Keilig, T.
Knieling, B.
Krabbe, A.
Schindler, K.
Schulz, B.
Wiedemann, M.
Zeile, O.
observatories!ground-based
organisations!NASA
archives!multiple!NASA/IPAC Infrared Science Archive (IRSA)
archives!individual!SOFIA
data!reduction
code!repository!Git
code!repository!GitHub
code!reviews
computer languages!Python
libraries!Astropy
libraries!numpy
libraries!scipy
packages!pandas
software!tools!PyQt
Astropy: ascl:1304.002
SAOImage DS9: ascl:0003.002

Maintaining SOFIA Redux

Benjamin Greiner, Aaron Bryant, Jonas Früh, Michael Hütwohl, Christof Iserlohe, Thomas Keilig, Bastian Knieling, Alfred Krabbe, Karsten Schindler, Bernhard Schulz, Manuel Wiedemann, and Oliver Zeile

SOFIA Data Center, Institute of Space Systems, University of Stuttgart, Germany; benjamin.greiner@irs.uni-stuttgart.de

Abstract.
The Stratospheric Observatory for Infrared Astronomy (SOFIA) has gathered a considerable amount of scientific data between first light in May 2010 and the final observing flight in September 2022. The joint mission by NASA and DLR produced a diverse set of astronomical data from several science instruments. While the mission was still active, the SOFIA Data Processing Software Team worked in cooperation with experts from the science instrument teams in order to write and maintain the reduction software suite "SOFIA Redux". The framework written in Python provides general purpose data reduction functions and dedicated modules for the SOFIA instruments EXES, FIFI-LS, FLITECAM, FORCAST, and HAWC+. It was continually improved and maintained until the end of the mission, with a most recent release of patch updates in August 2023. The SOFIA Data Center has forked the open source code base from GitHub and continues to keep the code fresh, accommodating API changes for underlying Python libraries. Furthermore, new features and improved reduction techniques are being implemented into the instrument reduction pipelines. We introduce the new home of the software at the SOFIA Data Center and encourage the astronomy community to collaborate in the software updates through discussions, bug reports, and code contributions.

1. Introduction

The Stratospheric Observatory for Infrared Astronomy (SOFIA) was a unique airborne observatory comprised of a 2.7 meter telescope integrated into a Boeing 747SP aircraft. The joint NASA and DLR project counted a total of 927 SOFIA flights between 2010 and 2022, of which 783 flights at least in part gathered scientific data observing the night sky in the visible to far-infrared spectrum.

SOFIA hosted a diverse set of science instruments, including cameras, spectrometers, and polarimeters. While each instrument team provided their own expertise about reducing the raw data, initially writing software in a variety of environments and languages, such as C, IDL, and Python, the SOFIA Data Processing Software (DPS) Team consolidated efforts to provide a common interface for data reduction. This resulted in the Python-based data reduction framework "SOFIA Redux", which provided general purpose data reduction functions and dedicated modules for the SOFIA instruments EXES, FIFI-LS, FLITECAM, FORCAST, and HAWC+ (Clarke et al. 2015).

During the active time of the mission the DPS worked in cooperation with experts from the science instrument teams in order to write and maintain Redux. The source

code has been publicly available on GitHub in a repository under the Universities Space Research Association (USRA) organization and releases were regularly published as the "sofia-redux" package on the Python Packaging Index (PyPI).

Data processing for reducing raw observation data has been mostly performed shortly after the observations with reduction pipeline versions current at that time. There was also some reprocessing with the most current version during the ramp-down phase of the mission between the last flight in September 2022 and the final shutdown of the SOFIA Science Center. The NASA/IPAC Infrared Science Archive (IRSA) now hosts the SOFIA data with the latest available products from the final reprocessing, but will not perform any further development of the reduction software.

The SOFIA Data Center (SDC) at the Institute of Space Systems (IRS) of the University of Stuttgart has secured funding in order to build a new science archive for the SOFIA data. It intends to reprocess the raw observation data with enhanced and updated reduction pipelines as they become available (Schulz et al. 2026). Improvements of the data quality are expected from new calibration references and enhanced reduction algorithm modules, especially for the FIFI-LS instrument (Bryant 2026).

2. Accommodating Dependency Drift

SOFIA Redux uses a number of libraries from the scientific Python stack such as NumPy (Harris et al. 2020), SciPy (Virtanen et al. 2020), pandas (The pandas development team 2026), and Matplotlib (Hunter 2007) as well as AstroPy (Astropy Collaboration 2013, 2018, 2022) throughout the algorithms.

As illustrated in Figure 1, the maintenance windows of many of these libraries have already slid past the SOFIA Redux v1.3.3 released from the SOFIA DPS in August 2023. Similarly, AstroPy makes the promise to maintain API stability within major releases for the period of one year but older versions will not get updates. This means that users installing the pipeline package from scratch and trying to perform their own data reduction either install outdated software into their environment or face incompatibilities with more recent software in the stack.

The SOFIA Data Center has updated all API usage in SOFIA Redux to accommodate current library versions. The existing unit tests have been most helpful with identifying deprecation warnings and errors with meaningful stack trace messages. The continiuous integration (CI) workflows regularly run these tests during further development and will ensure that the code base remains within the supported API surface of the dependencies.

Further changes include the migration from the end-of-life Qt5 framework using PyQt5 to the current Qt6 using PyQt6 [1] , and the integration of SAMP communication with SAOImage DS9 through ds9samp [2] instead of the formerly used but unmaintained pyds9 via XPA.

[1] PyQt: https://www.riverbankcomputing.com/software/pyqt/
[2] ds9samp: https://github.com/cxcsds/ds9samp

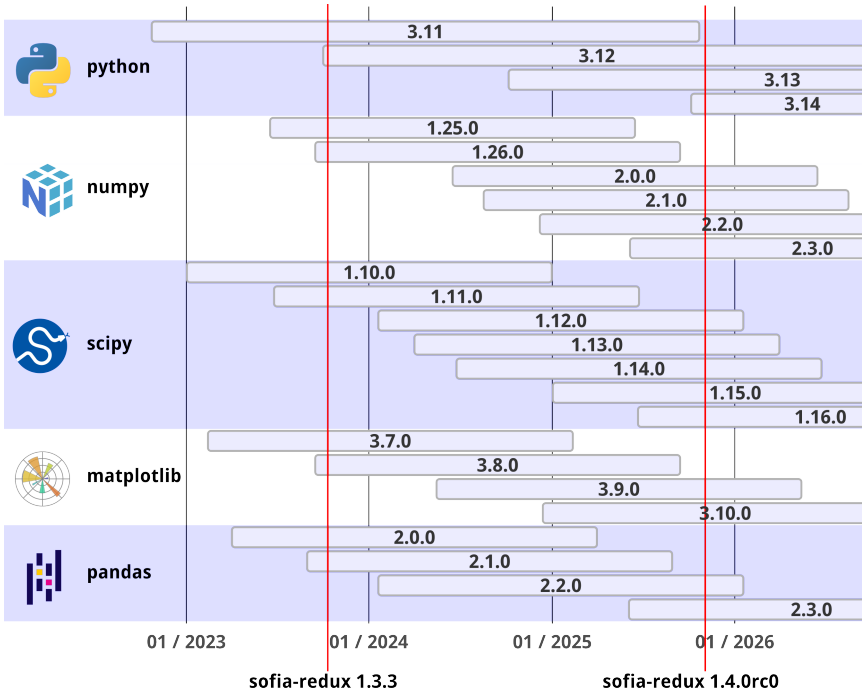


Figure 1. Software maintenance windows of Python releases and libraries from the scientific Python stack used in SOFIA Redux. Illustration based on SPEC-0 (https://scientific-python.org/specs/spec-0000/)

3. Development Workflow

The SOFIA Data Center has forked the open source code base from the USRA GitHub repository and maintains a repository on an internal GitHub Enterprise instance of the University of Stuttgart. The repository is browsable with an account from the University and guest accounts are available for collaborators if needed.

The development workflow uses feature branches and pull requests (PR) for code reviews before merging into the main *sdc-dev* branch. Since taking over the code base, the SDC team has pushed more than 200 individual commits and merged more than 50 pull requests into the main branch.

The main branch, whenever updated, is regularly pushed to the publicly available GitHub.com repository [3]. Outside collaborators without a university guest account are welcome to fork from the public repository and contribute through issues and pull requests.

4. Publication

The SDC has received ownership of the original and established *sofia-redux* package on the Python Packaging Index (PyPI), which enables users to migrate to the updated

[3] Public SOFIA Redux repository: https://github.com/SOFIA-Data-Center/sofia_redux

versions seamlessly. Just before presentation at the ADASS 2025 conference, a release candidate in version 1.4.0rc0 was published and a final release of SOFIA Redux 1.4.0 was published shortly after (Greiner et al. 2025). The same tags and releases were also created in the GitHub hosted repository for reference.

The documentation for SOFIA Redux is maintained in the same repository as the python source code using Sphinx and is currently published on a GitHub pages service of the University of Stuttgart GitHub Enterprise instance [4] . As the SDC project time is limited, and thus the availability under the uni-stuttgart.de URLs may not remain permanent, the SDC is in the process of obtaining a top level DNS domain in order to provide a more stable URL for the SOFIA archive and its related services, including the Redux documentation.

5. Conclusion

The SOFIA Data Center at the University of Stuttgart continues to maintain and further develop the SOFIA Redux data reduction framework. The software has been updated to accommodate recent versions of the scientific Python stack and other dependencies. New features and improved reduction techniques are being implemented into the instrument reduction pipelines, especially for the FIFI-LS instrument. The SDC encourages the astronomy community to collaborate in the software updates through discussions, bug reports, and code contributions.

Acknowledgments. The SDC is funded by the DLR Space Administration under grant number FKZ 50OK2404. The SDC would like to thank the dedicated members of the SOFIA team as well as all the science instrument teams for creating an excellent data and software foundation. Melanie Clarke, former development lead of the DPS and Anastasia Laity from IPAC have been most helpful in providing ownership rights to the *sofia-redux* package on PyPI.

References

Astropy Collaboration 2013, A&A, 558, A33. eprint: 1307.6212
— 2018, AJ, 156, 123. eprint: 1801.02634
— 2022, ApJ, 935, 167. eprint: 2206.14220
Bryant, A. 2026, in ADASS XXXV, edited by K. Polsterer, & S. Wagner (San Francisco: ASP), vol. TBD of ASP Conf. Ser., 999 TBD
Clarke, M., Vacca, W. D., & Shuping, R. Y. 2015, in ADASS XXIV, edited by A. R. Taylor, & E. Rosolowsky (San Francisco: ASP), vol. 495 of ASP Conf. Ser., 355
Greiner, B., Knieling, B., Bryant, A., Früh, J., Fischer, N., & Valek, R. 2025, Sofia-data-center/sofia_redux. URL https://doi.org/10.5281/zenodo.17980055
Harris, C. R., et al. 2020, Nature, 585, 357. URL https://doi.org/10.1038/s41586-020-2649-2
Hunter, J. D. 2007, Computing in Science & Engineering, 9, 90
Schulz, B., et al. 2026, in ADASS XXXV, edited by K. Polsterer, & S. Wagner (San Francisco: ASP), vol. TBD of ASP Conf. Ser., 999 TBD
The pandas development team 2026, pandas-dev/pandas: Pandas. URL https://doi.org/10.5281/zenodo.18328522
Virtanen, P., et al. 2020, Nature Methods, 17, 261

[4] https://pages.github.tik.uni-stuttgart.de/IRS-SOFIA-Data-Center/sofia_redux/

Astronomical Data Analysis Software and Systems XXXV
*ASP Conference Series, Vol. **VOL** Compiled: Tuesday 30th June, 2026@ 01:30*
K. Polsterer, D. Bohlender, S. Gwyn, A. Hincks and P. Teuben, eds.
*© **YEAR** Astronomical Society of the Pacific*

ZZZDEBUG!254.tex
Borra
Sagar, B. S.|textbf
Buchholz Elsa, B. E.
Canbolat Utku, C. U.
Desing Raja
Chithralekha, R. C.
Drobek Mark, D. M.
Haupt Lars, H. L.
Heßling Hermann, H. H.
High Performance
Computing!hybrid
infrastructure!ADASS
2025
Cloud Computing!hybrid
infrastructure!ADASS
2025
Hybrid
infrastructure!kubernete
and slurm!ADASS 2025

Hybrid Cloud HPC Infrastructure @ DZA

Lars Haupt,[1] Sagar Borra,[1] Elsa Buchholz,[1] Ahmet Utku Canbolat,[1] Chithralekha Desing Raja,[1] Marc Drobek,[1] Hermann Heßling[1], and LZR Team[2]

[1]*Deutsches Zentrum für Astrophysik, Görlitz, Saxony, Germany;*
sagar.borra@dzastro.de elsa.buchholz@dzastro.de utku.canbolat@dzastro.de chithralekha.desing_raja@dzastro.de marc.drobek@dzastro.de lars.haupt@dzastro.de hermann.hessling@dzastro.de

[2]*Center for Information Services and High Performance Computing (ZIH) at TU Dresden, Dresden, Saxony, Germany*

Abstract. Traditionally, the majority of scientific computing centers have focused solely on high-performance computing (HPC) with a strong basis on providing software and hardware for high-intense and computationally expensive scientific algorithms. The cornerstone of HPC software (e.g., Slurm) need to fulfill a variety of requirements, such as hardware resources sharing, dynamic scheduling and queuing, adaptive provisioning of resources, limitation of requested resources, etc. However, the advent of cloud computing has introduced new software solutions (e.g., Kubernetes), which in general enable different operational concepts for such scientific workloads. In this paper, we aim to highlight the differences of both concepts and show a path forward for the German Centre for Astrophysics (DZA) in employing both solutions.

1. Introduction

Astronomy and astrophysics are currently experiencing a new golden age, with a multitude of new discoveries being made, particularly through multi-messenger astronomy. The combined evaluation of multiple messengers leads to an even greater increase in data volume. In order to analyze this data as efficiently and quickly as possible, efficient and easy-to-use pipelines are needed. It must be ensured that data engineers and astrophysicists are not distracted from their actual research by unnecessarily complicated handling of these pipelines. Ideally, tasks such as installation, setup, and deployment should be kept completely separate from the scientists, and all calculations should run on the best possible and most suitable hardware.

The analysis of huge amounts of data must follow the "software to the data" paradigm. To facilitate the development of workflows that take this into account, interactive testing with data in remote data centers should be easy to perform, see the solution based on Jupyter notebook presented in Trattner (2024) .

Conventional approaches such as classic high-performance computing (HPC) need to be rethought and made significantly more user-friendly. However, simply moving to the cloud, which appears to solve many of the problems, is short-sighted and generates new or different problems. A good combination of both worlds seems appropriate here.

The German Center for Astrophysics (DZA (2026)) is currently in the build-up phase. A key advantage of a new large-scale research center is that existing procedures and processes do not need to be taken into account, meaning that forward-looking structures and processes can be set up and established from the outset.

This article outlines the current status of the hybrid cloud HPC infrastructure being set up at the DZA to handle the data streams that need to be analyzed and managed as part of its participation in numerous large-scale astronomical projects.

2. Cloud vs. High Performance Computing

The advantages and disadvantages of the HPC and cloud computing approaches can be assessed pragmatically using the two most established management tools: Slurm for HPC workload management and Kubernetes for cloud automatic deployment. Table 1 shows a comparison of Slurm and Kubernetes based on eight categories.

Table 1. Slurm and Kubernetes comparison

Category	Slurm (HPC)	Kubernetes (Cloud)
Target use-case	Heavy, but "short-running" batch compute	Never failling, always up-to-date end-user service
High Availability	Plattform: not required Jobs: not required	Plattform: required (9er rules) Services: required (9er rules)
Environment	Self-managed terminal, env-script singularity Containers	Entirely image/container based
Automation	Mostly manual job triggers (can be scripted though)	GitOps oriented fully automated approach
Queuing/Batching	Native	Scheduling only (but Batching can be added)
Backup/Recovery	Input/Result data: required Jobs: not required	Required for all services
Resource Sharing Model	Between all authorized parties (reservation possible)	Between all authorized parties
Main Adoption	Academia / Research	Industry

The above comparison shows that in most categories, the two worlds are diametrically opposed. If a hybrid, unifying cloud HPC infrastructure can be established, it would theoretically be possible to deploy highly available applications interactively and without delay on the hardware best suited to the use case in question. The goal at the DZA is to put this into practice.

3. Unleashing the potential of hybrid cloud and high performance computing

The potential advantages of a hybrid cloud HPC infrastructure well known, which is why several development and research teams have been working on ways to connect

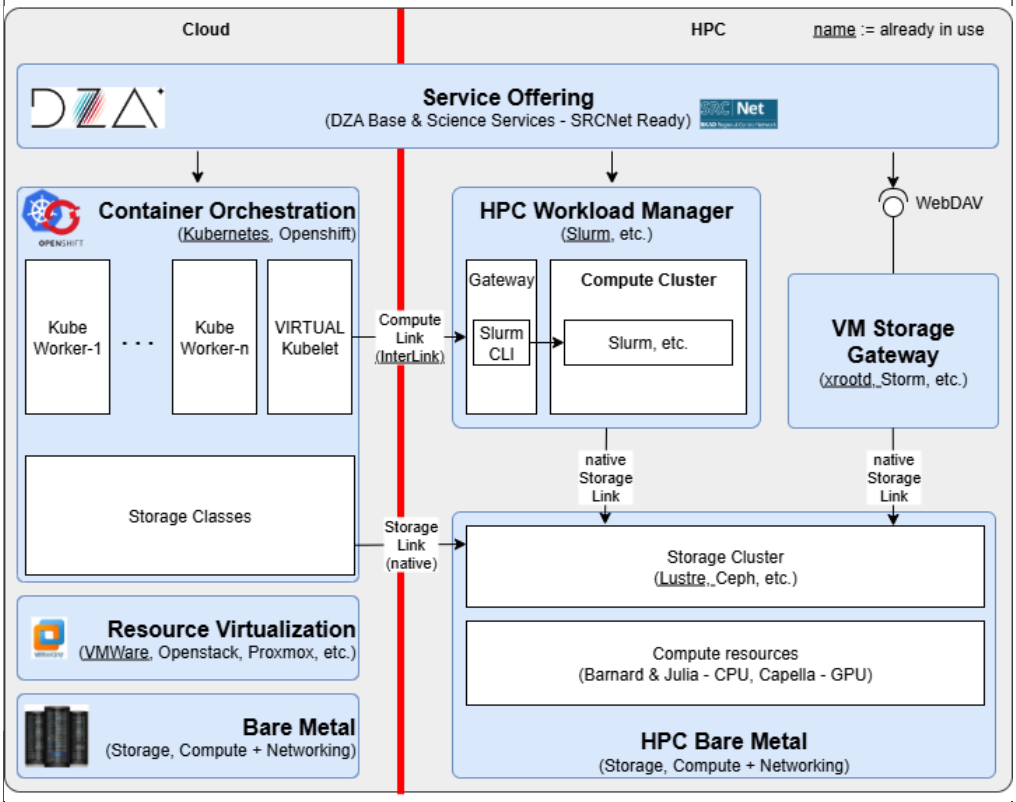


Figure 1. Schematic representation of the IT system architecture of the DZA

the compute plane of Kubernetes and Slurm over the last two to three years. Solutions such as InterLink (2026) and Slinky (2026) are the decisive factors here and also form the core of the DZA IT system architecture shown in figure 1, with the current focus on interLink (see "Compute Link" in fig. 1). The specific goal is to transfer or schedule pods or jobs in Kubernetes and then automatically place them in a suitable queue as jobs within Slurm. The progress made so far is promising. Due to the early development status of interLink, it is not yet possible to assess whether this approach covers all use cases. In particular, the performance and manageability of complicated scenarios still needs to be evaluated.

In addition to the computational level, connecting Kubernetes and Slurm environments at the data plane (see "Storage Link" in fig. 1) is essential. While Kubernetes focuses on StorageClasses for block storage and object storage (but is not limited to these), Slurm is typically connected to traditional file storage (Lustre and similar). If it is absolutely necessary to enable access to the same data for both Kubernetes services and Slurm computing jobs, the main question will focus on finding a fast and reliable way to access data without having to copy it. The identification and evaluation of suitable solutions in this regard is still ongoing.

In addition to the hybrid cloud HPC approach, the DZA is working on placing services and data intuitively and interactively on the HPC systems so that users are spared the established, somewhat cumbersome route via SSH tunnels and scripts. Interest-

ing solutions are offered here by XRootD (2026) and Storm as well as Teapot (2026) from the InterTwin project environment. In particular, the results already achieved with XRootD using the WebDAV interface are very promising.

Regardless of the solutions applied at the software level, a functional cloud science platform also requires sufficiently powerful hardware. This must ensure that resource virtualization on the cloud side, data-intensive computing and data archiving, as well as high and fast data availability on the HPC side are guaranteed. Currently, the DZA has the Barnard CPU cluster (2496 Intel CPUs), the Capella GPU cluster (60 NVIDIA H100 GPUs for computing), and a shared memory system with 48 TB of RAM available for computing. All computing systems are connected to a fast InfiniBand network and have direct access to fast disk storage called Narwhal (10 PB). In addition, slow tape storage (10 PB), which is connected via Fibre Channel, is provided for archives and catalogue data. The Kubernetes cluster, managed via VMware, is based on AMD EPYC 7003 CPUs (64 cores, 32 DIMMs, 12 NVMe). To ensure that scientists can not only submit jobs as easily as possible, but also in sufficient numbers, the available hardware capacity at the DZA is constantly being expanded.

4. Conclusions and Future View

The current development status of the hybrid cloud HPC infrastructure being set up at the DZA is promising. In particular, the use of the "interLink" tool from the interTwin (2026) project for direct coupling of the cloud and HPC worlds, as well as the use of XRootD for interactive data transfer via WebDAV, already offers significant advantages over pure HPC or cloud approaches. However, it remains to be seen what previously unconsidered difficulties will arise. In the near future, the authors will focus in particular on evaluating the performance of the proposed solutions, which plays an essential role for application in large-scale astronomical projects. In addition, further alternative solutions will be considered and evaluated.

Regardless of this, another major challenge in the near future is the further expansion of the DZA hardware. Here, it is necessary to select suitable network and storage technologies, procure sufficient capacity, and optimally integrate them with each other.

Acknowledgments. The authors would like to thank the administrator team at the Lehmann Computer Center at TU Dresden for their strong support in setting up the DZA infrastructure and services.

References

DZA 2026, Deutsches Zentrum für Astrophysik. URL https://www.deutscheszentrumastrophysik.de/en/research

InterLink 2026. URL https://github.com/interlink-hq/interLink

interTwin 2026. URL https://www.egi.eu/project/intertwin/

Slinky 2026. URL https://github.com/slinkyproject

Teapot 2026. URL https://github.com/interTwin-eu/teapot

Trattner, M. 2024, Software to the data: Running containers interactively in computing centres. Technical report, URL https://www.punch4nfdi.de/sites/sites_custom/site_punch4nfdi/content/e218451/e218467/e218469/e274306/e274309/Software_to_the_data__running_containers_interactively_in_computing_centres.pdf

XRootD 2026. URL https://xrootd.org/

Astronomical Data Analysis Software and Systems XXXV
*ASP Conference Series, Vol. **VOL** Compiled: Tuesday 30th June, 2026@ 01:30*
K. Polsterer, D. Bohlender, S. Gwyn, A. Hincks and P. Teuben, eds.
*© **YEAR** Astronomical Society of the Pacific*

ZZZDEBUG!255.tex
Savalle, R.|textbf
Godard, J.-P.
Cavadore, C.
Valls-Gabaud, D.
FOOBAR!conference!ADASS 2019
FOOBAR!organisations!ASP
FOOBAR! ascl:1101.010

WIVONA - Opening the Virtual Observatory to the Amateur Astronomical Community

R. Savalle,[1] Jean-Paul Godard,[2] Cyril Cavadore,[3] and David Valls-Gabaud[4]

[1]*DIO/Paris Astronomical Data Centre (PADC), Observatoire de Paris, UAR2201-CNRS, PSL Research University., 5, place Jules Janssen, 92195 Meudon, France;* *renaud.savalle@obspm.fr*

[2]*Observatoire de Gravelle, 94410 Saint Maurice, France*

[3]*Alcor-System, 42570 Saint Heand, France*

[4]*LUX, Observatoire de Paris, CNRS, PSL, 61 Avenue de l'Observatoire, 75014 Paris, France*

Abstract. Within the framework of Open Science and Citizen Science, the WIVONA (We Implement Virtual Observatory Needs of Amateur Astronomers) project, funded by the Gemini Pro/Am initiative at Observatoire de Paris, aims to promote software interoperability as well as to provide access to astronomical data through the Virtual Observatory (VO) for the amateur astronomical community. Over the past couple of years, we have implemented several IVOA tools and protocols in the popular PRISM v11 software, an all-in-one observatory management solution. First, we have developed a native Delphi implementation of SAMP (Simple Application Messaging Protocol) to exchange celestial coordinates and images between PRISM and VO software (such as Aladin). Then, by embedding a Python console within PRISM, we have enabled the use of Python scripts for VO discovery and data retrieval with the PyVO, AstroPy and Astroquery packages. We have also implemented access to other services providing an open API such as Astro-Colibri (a transient event broker that allows filtering and selection of interesting targets for follow-ups), the Exo-Clock initiative (which facilitates the observations of exoplanet transits), and the BHTOM project (Black Hole Target Observation Manager: a web platform designed to coordinate a telescope network). These developments pave the way for an integrated workflow for processing and archiving observational results. The development of our VO interfaces greatly helps the amateur community to access professional databases and tools, enabling them to significantly enhance their contributions to astronomical research.

1. Astro-Colibri Interface

We have incorporated in PRISM stellar and extra-galactic transients, as reported by Astro-Colibri (Reichherzer et al. 2024) [1], and now observed by amateur astronomers within the French Pro/Am RAPAS network. To ensure a fast, user-friendly interface, we implemented a two-click procedure in the new VO-compliant version of PRISM.

[1]PRISM v11 is available at https://www.prism-astro.com and see https://astro-colibri.com/

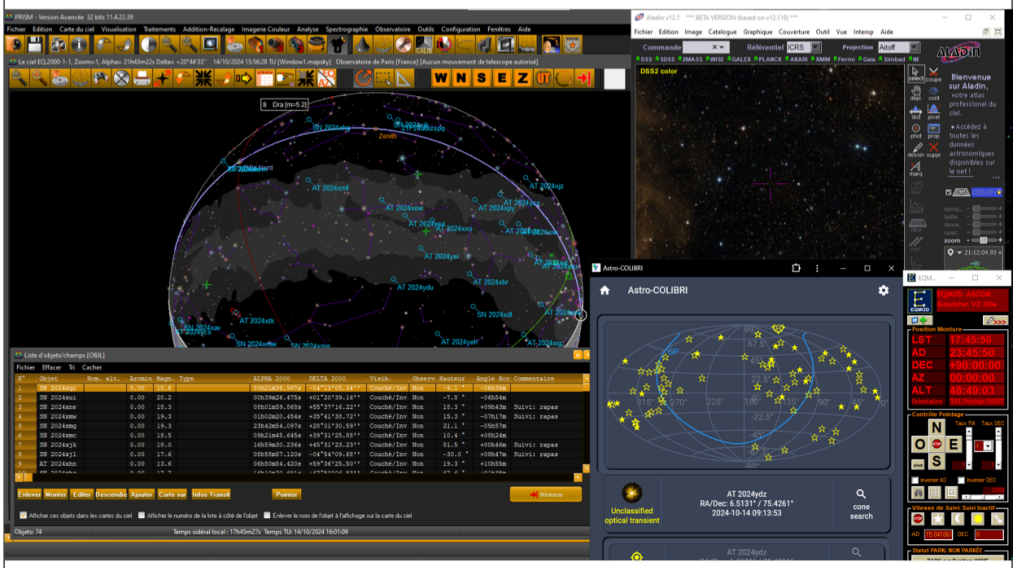


Figure 1. Skymap produced in PRISM with the latest transients from Astro-COLIBRI, as implemented by our interface.

The first click on the PRISM screen requests latest transients from Astro-COLIBRI and plots them on a sky map. A second click selects the object to be observed and automatically points the telescope to that position, see Fig. 1.

2. Python Console and Editor

A python scripting facility has been integrated within PRISM using Python4Delphi so that the user can load Python scripts, edit, parametrise and execute them. Our software automatically detects the Python platform installed on the computer. Once the user installs the packages `astropy`, `astroquery` and `pyvo` using `pip`, they are available in PRISM. Scripts can easily be debugged locally during the observations thanks to the text editor next to the Python console, see Fig. 2. Using VO queries, amateurs can access existing archives and measurements, and prepare the list of objects to observe. Furthermore, we are developing a Python interface to the connected hardware devices so the user will be able to drive the Telescope mount, control the Cameras, etc. allowing full automatization of the observing process with Python.

3. BHTOM Interface

Our latest development concerns BHTOM (Black Hole Target Observation Manager Wyrzykowski et al. 2024), which aims at enabling seamless collaboration and efficient data management between professional and amateur astronomers. It features a powerful form interface to query targets for which follow-up observations are requested by researchers. These targets are also available through an API, which we use to build a form to extract the objects of interest in PRISM. This allows the amateur astronomer

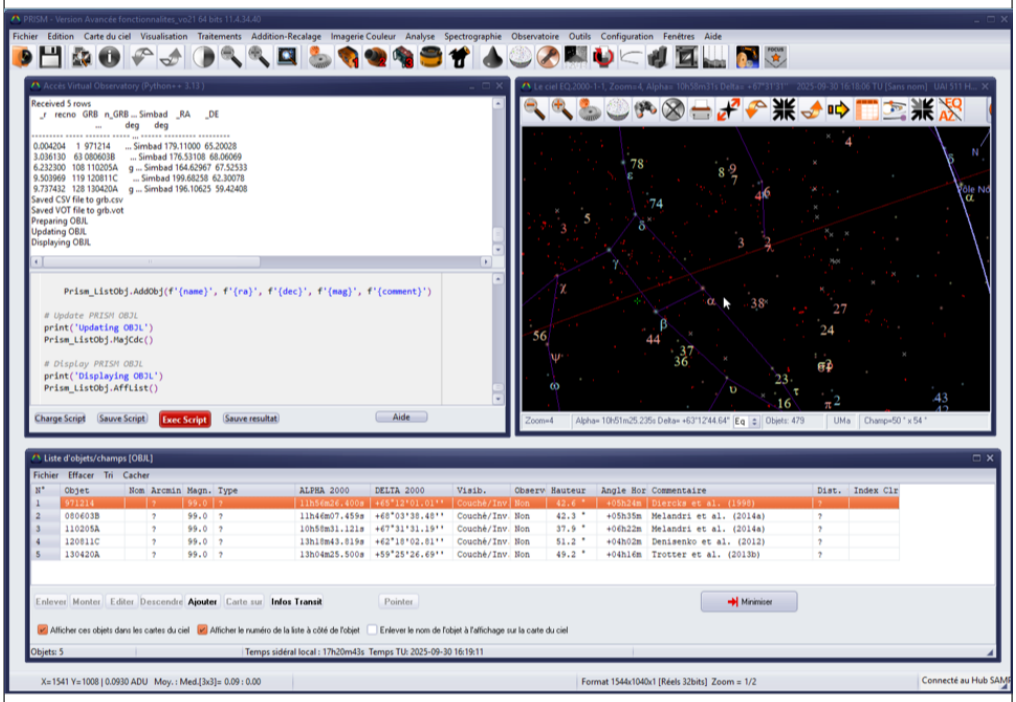


Figure 2. The python console and editor.

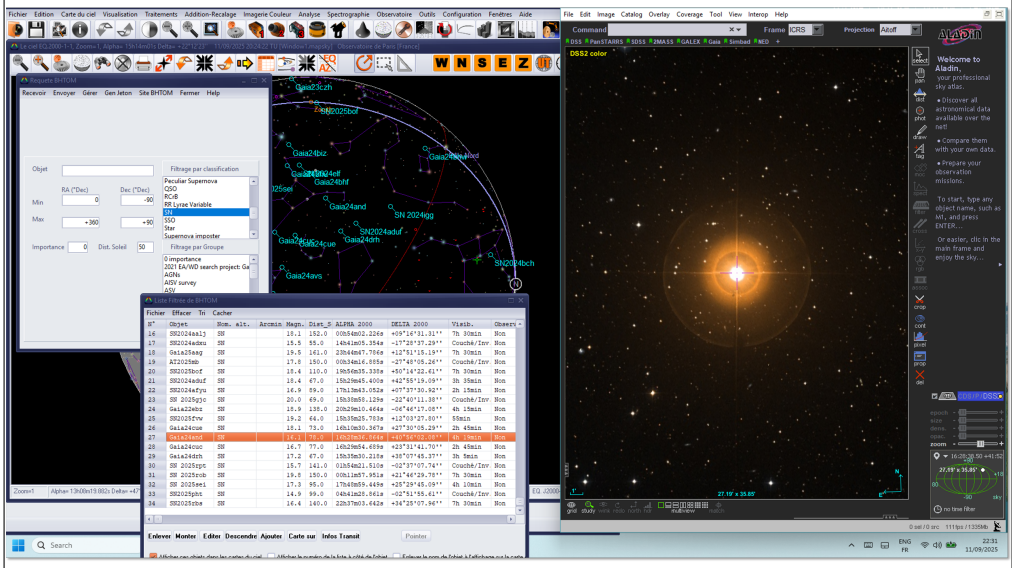


Figure 3. The Black Hole Target Observation Manager form and target list.

to use a simple interface to ensure a very efficient and quick access to the targets to be observed, leaving the nitty-gritty aspects to the underlying code, as shown in Fig. 3.

4. Current Status And Further Details

These three recent applications follow a number of useful features implemented over the past few years (Savalle et al. 2024, 2025) and detailed at these meetings:

- GEMINI workshop at SF2A (June 2024, Savalle et al. 2024)
- Photometry school at Marseille (June 2024)[2]
- Third Astro-Colibri workshop at CEA/IRFU, September 2024), see https://indico.ijclab.in2p3.fr/event/10545/contributions/ [3]
- User documentation at amateur forums[4]
- Two workshops at Rencontres du Ciel et de l'Espace RCE (AFA) in November 2024, https://www.afastronomie.fr/forums
- RAPAS workshop at Meudon Observatory (December 2024) https://rapas.imcce.fr/RAPAS-atelier3-prelim.html
- GEMINI workshop at the SF2A conference (June 2025) https://ui.adsabs.harvard.edu/abs/2025sf2a.conf..241S/abstract
- ACME Multimessenger Citizen Science: Training Event for Amateur Astronomers[5]
- Fourth Astro-Colibri workshop at CEA/IRFU (October 2025), https://indico.ijclab.in2p3.fr/event/11610/overview and https://www.youtube.com/@astro-colibri/videos
- API ProAm workshop 2025 at Observatoire de Paris (November 2025), https://gemini.obspm.fr/journees-api-proam-2025/

Acknowledgments. We gratefully acknowledge support from the 2024 and 2025 API Pro/AM programmes at Observatoire de Paris.

References

Reichherzer, P., Schussler, F., Lefranc, V., Alkan, A., & Tjus, J. B. 2024, in Astromical Data Analysis Software and Systems XXXI, edited by B. V. Hugo, R. Van Rooyen, & O. M. Smirnov, vol. 535 of Astronomical Society of the Pacific Conference Series, 159

Savalle, R., Godard, J.-P., Cavadore, C., & Valls-Gabaud, D. 2024, in Proceedings of the Annual meeting of the SF2A, edited by M. Béthermin et al., 497

— 2025, in Proceedings of the Annual meeting of the SF2A, edited by A. Siebert et al., 241

Wyrzykowski, L., Zielinski, P., Mikolajczyk, P., & Kotysz, K. 2024, in EAS2024, European Astronomical Society Annual Meeting, 2399

[2] https://gemini.obspm.fr/2024-edp/

[3] https://www.youtube.com/watch?v=zWUI6d9sCOs

[4] https://forum.prism-astro.com/viewforum.php?f=34 and https://forums.sharpcap.co.uk/viewtopic.php?p=42943

[5] June 2025, see https://citizen-science.sciencesconf.org/data/pages/2025_06_10_ACME_WIVONA_VO_python_integration.pdf and https://www.youtube.com/watch?v=nElUumgb_Mg

Astronomical Data Analysis Software and Systems XXXV
*ASP Conference Series, Vol. **VOL** Compiled: Tuesday 30th June, 2026@ 01:30*
K. Polsterer, D. Bohlender, S. Gwyn, A. Hincks and P. Teuben, eds.
*© **YEAR** Astronomical Society of the Pacific*

ZZZDEBUG!256.tex
Smith, L. C.|textbf
Hodgkin, S.
de Angeli, F.
Ahmed, S.
Busso, G.
Ford, D.
Harrison, D.
Irwin, J.
Murphy, D.
Rixon, G.
Walton, N.
algorithm!detection
observatories!space-based!
computing!GPU
astronomy!exoplanets
observatories!space-based!
stars!light curves
organisations!ESA
astronomy!exoplanet
transits
exoplanet!
ascl:1910.005
PLATO! ascl:1506.003
Transit! ascl:1704.008
GAIA! ascl:1403.024

transitPipe: a key component of the ESA PLATO Exoplanet Analysis System

Leigh C. Smith,* Simon Hodgkin, Francesca de Angeli, Saad Ahmed, Giorgia Busso, Dominic Ford, Diana Harrison, Jonathan Irwin, David Murphy, Guy Rixon, and Nicholas Walton

Institute of Astronomy, University of Cambridge, Madingley Rd, Cambridge CB3 0HA, UK

**lsmith@ast.cam.ac.uk*

Abstract. We outline the current status of the systems we are designing to detect and characterise exoplanet candidates with PLATO light curve data in the early stages of processing (transitPipe). The results of transitPipe comprise a list of vetted and graded exoplanet candidates which will be shared with the community, and passed on to analysis at the next stage (planetPipe). Our process includes robust strategies for filtering stellar variability, and a new GPU-based detection algorithm for the transit signals: CETRA (Cambridge Exoplanet Transit Recovery Algorithm). We also summarise the vetting and grading processes to be applied to all candidates, and show how the PLATO multi-camera data gives us extra leverage for the identification of false positive signals for blended systems.

1. Introduction

ESA's PLATO mission is designed to find and characterize exoplanets through near-continuous observation of a few hundred thousand stars, with a particular focus on terrestrial planets orbiting bright, nearby hosts. By combining high-precision photometry with long temporal baselines, PLATO will detect planetary transits while simultaneously measuring stellar oscillations, enabling accurate determination of stellar and planetary properties (Rauer et al. 2025). Achieving these science goals requires a highly scalable and robust data processing pipeline capable of handling unprecedented data volumes and cadence. The PLATO Exoplanet Analysis System (EAS) has been designed as a modular framework, within which transitPipe plays a critical role in the identification, vetting, and prioritisation of exoplanet candidates for further analysis.

2. PLATO pipeline - filtering & detection

All PLATO light curves will have instrumental systematics removed with PDC-MAP or Republic (Barragán et al. 2024), and stellar variability and planet signatures will be preserved. These combined Level 1 PLATO light curves are fed to transitPipe, and will see a single variability filtering algorithm and a single transit detection algorithm. Extra combinations may be run for subsets of the data, e.g. variable Young Stars. In addition, simulated data will be passed through the same system to help identify and characterise

true and false positives. This approach will help us to estimate planet completeness later in the mission.

For transit detection we have developed a new a code, CETRA, written using the CUDA framework for NVIDIA GPUs. CETRA separates the task into a search for transit signals across linear time-space, followed by a phase-folding of the former to enable a periodic signal search, using a physically motivated transit model to improve detection sensitivity. CETRA is up to a few orders of magnitude faster for high cadence light curves (see Fig. 1). This is a factory-scale approach developed at Cambridge that enables us to handle the datascale that PLATO will provide.

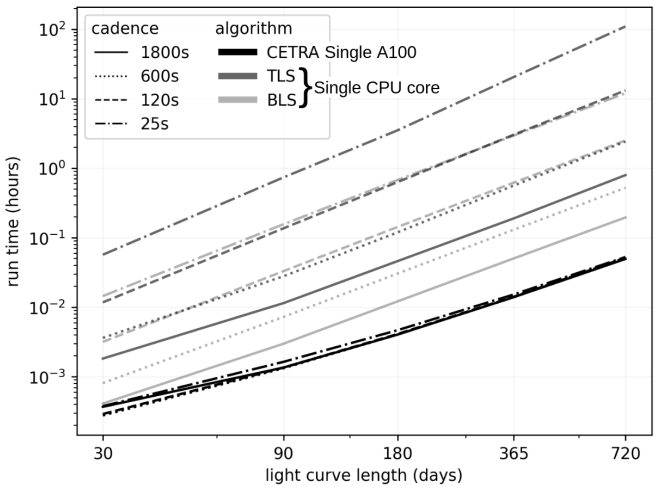


Figure 1. CETRA transit search times versus typical CPU based alternatives. Adapted from Smith et al. (2025).

2.1. Testing with TESS data - the HD 101581 system

To demonstrate the CETRA transit detection software Smith et al. (2025) successfully recovered the signals of the three Earth-sized planets in the HD 101581 (TOI-6276) system within the TESS high cadence light curves. Their fig. 6 shows the result of the initial search for singular transit-like signals in linear time vs. transit duration space. The TESS periodic transit signals are shown in their fig. 8. Two of the planets were previously detected with TLS and validated in the lower cadence light curves by Kunimoto et al. (2025), but the third was found to be too low SNR to statistically validate.

CETRA found improved signal-to-noise ratios in the short cadence data. The signal of the previously unvalidated planet (TOI-6276.03) in particular was boosted from SNR = 7.9 to SNR = 16.0, which is potentially sufficient to reconsider the statistical validity of the planet.

3. Vetting PLATO planetary candidates

Vetting is another major component of the transitPipe suite. Astrophysical and instrumental signals and noise sources that resemble transits are typically more common than

real transiting planetary signals. Consequently, candidates identified by transit detection algorithms must be carefully vetted.

3.1. PLATO target samples

The PLATO First Long Pointing (LOPS2) is described by Nascimbeni et al. (2025). It will make up the first two years of observations, and is separated into four subsets: The P1 sample is 8235 targets, and is comprised of FGK dwarfs and subgiants with NSR $\leq$ 50 ppm in one hour and $V \leq 11$; The P2 sample is 699 targets, and is comprised of FGK dwarfs and subgiants with NSR $\leq$ 50 ppm in one hour and $V \leq 8.5$; The P4 sample is 12,415 targets, and is comprised of M dwarfs with $V \leq 16$; And the P5 sample is 167,149 targets, and is comprised of FGK dwarfs and subgiants with $V \leq 13$. For the P1, P2 and P4 samples, imagettes will be downlinked from the satellite, meaning we will obtain PSF fitted light curves for targets and their contaminants ($\leq$ 10 per target). However, for the P5 sample we will largely only have aperture data, and consequently, vetting of candidate planets from these targets presents a greater challenge.

3.2. Vetting the P5 Sample

PLATO has two major advantages with respect to previous transit detection missions: Firstly, the Gaia catalogue provides information on potential sources of contamination; And secondly, PLATO is armed with multiple cameras. On the latter point, figure 1 of Nascimbeni et al. (2025) shows the geometry of the PLATO field of view, with the number of "normal" cameras covering a given line of sight colour coded. Field of view regions covering 325/153/847/824 deg^2 are observed with 24/18/12/6 cameras, respectively.

Gaia provides us with precise positions of sources in the field (including targets and their contaminants), and the CCD pixel grid is expected to differ between cameras (see Fig. 2). The level of the contaminating flux in each camera's light curve will therefore be different. If the dip in the target light curve is due to a contaminant, the depth of the transit will be inconsistent between cameras. A statistical test of the distribution of depths will reveal if the signal originates at the target, in which case the distribution will be normal; or a contaminant, in which case the distribution should be significantly non-normal.

4. Summary

This paper describes transitPipe, the early-stage exoplanet detection component of the ESA PLATO Exoplanet Analysis System. transitPipe processes PLATO light curves to filter stellar variability, detect transit signals, and produce a vetted and graded list of exoplanet candidates for later analysis. A key innovation is CETRA, a GPU-accelerated transit detection algorithm that enables factory-scale searches of high-cadence data with improved speed and sensitivity relative to traditional CPU-based methods. The pipeline also incorporates robust vetting strategies, including the use of Gaia data and PLATO's multi-camera observations to identify false positives, particularly for blended systems. Overall, transitPipe is designed to efficiently handle the data volume expected of PLATO, while maximizing completeness and reliability of detected exoplanet candidates.

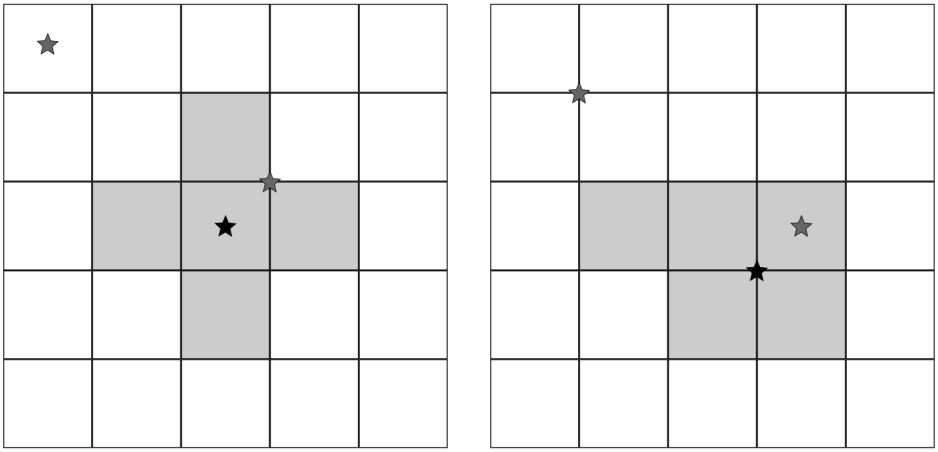


Figure 2. Example of camera-specific target location (black star), contaminant locations (grey stars), and aperture mask (light grey pixels). Differing transit depths between cameras can indicate where a signal is induced by a contaminant.

References

Barragán, O., Aigrain, S., & McCormac, J. 2024, RAS Techniques and Instruments, 3, 198. eprint: 2404.06132
Kunimoto, M., et al. 2025, AJ, 169, 47. eprint: 2412.08863
Nascimbeni, V., et al. 2025, A&A, 694, A313. eprint: 2501.07687
Rauer, H., et al. 2025, Experimental Astronomy, 59, 26. eprint: 2406.05447
Smith, L. C., et al. 2025, MNRAS, 539, 297. eprint: 2503.20875

Astronomical Data Analysis Software and Systems XXXV
*ASP Conference Series, Vol. **VOL** Compiled: Tuesday 30th June, 2026@ 01:30*
K. Polsterer, D. Bohlender, S. Gwyn, A. Hincks and P. Teuben, eds.
*© **YEAR** Astronomical Society of the Pacific*

KM3NeT
LLM
ZZZDEBUG!257.tex
Schnabel, J.|textbf

Building scientific practices with LLMs in KM3NeT

Jutta Schnabel[1] *for the KM3NeT Collaboration*

[1]*Erlangen Centre for Astroparticle Physics, Friedrich-Alexander Universität Erlangen-Nürnberg, Erlangen, Germany; jutta.schnabel@fau.de*

Abstract. With the emergence of Large Language Models (LLMs), the accessibility to information encoded in text for scientific purposes has significantly increased and provides a source for the augmentation of scientific practices within physics collaborations and for open science. In this contribution, the interface generation to internal and external sources of scientific information and application to enhance documentation, software quality practices and support the building of scientific pipelines in the KM3NeT collaboration is shown.

1. Epistemic layers of scientific knowledge

1.1. From Context to Data

Scientific model building and verification is a multi-layered process that culminates in the generation of data. This data serves as the fundamental currency of modern science; however, its accessibility and perceived value are deeply intertwined with the cultural, social, scientific, and technological contexts of both the producers and the users. Initiatives toward the FAIRification of data address the *technological level*—ensuring that data is made accessible and its utility enabled so that other researchers are able to reproduce results.

To this end, the underlying scientific models must be understood and the *ontological model*, in which this data gains its meaning, must be clearly defined. Although FAIRification addresses this ontological level through its dimension of reusability, a comprehensive understanding of the full context of data production and analysis necessitates fundamental knowledge of both the specific scientific domain and the technological methods concerned.

Consequently, science frequently remains an endeavor of a specialized few, predicated on the assumption of a fully qualified (typically English-speaking) researcher possessing domain-specific knowledge and competence in the use of specialized tools. To attain that expertise, the researcher must navigate the *contextual level*, acquiring the foundational scientific concepts, tool proficiencies, and an understanding of the research questions that underpin the field.

1.2. Large Language Models as Epistemic Engines

Across all three levels, advancements in natural language processing offer the opportunity to leverage the capabilities of Large Language Models (LLMs) to transform the linguistic representation of information, thereby enhancing access to scientific knowl-

edge. Adapting these models to specific tasks that support the interpretation of scientific analyses promises improvements across each epistemic level. However, their inherent nature as probabilistic machines prone to hallucinations necessitates the implementation of rigorous verification methods and safeguards to ensure scientific accuracy.

By utilizing pre-trained models alongside an expanding ecosystem of tools to create LLM agents, specialized systems addressing each epistemic level can be developed. Most prominently, through prompt engineering, specific tasks or contexts can be inferred; fine-tuning allows for a more detailed representation of niche epistemic fields; and retrieval-augmented generation (RAG) can anchor a model to authoritative, domain-specific sources of information.
At the *contextual level*, the capability to provide abstractions and summaries of content helps bridge the knowledge gap for non-expert users exploring a scientific domain, while multilingual support renders previously English-only research more accessible. At the *ontological level*, the comprehension of exact methodologies or coding environments can be supported by making obscure documentation more discoverable, providing background on mathematical or scientific concepts, and assisting in the development and vetting of scientific code. At the *technological level*, semantic search facilitates the retrieval of related studies and datasets from established catalogs and can provide supplemental metadata for multi-modal information, such as graphics and tables.

2. The LLM toolbox in KM3NeT

In order to support the above multitude of functionalities, access to data and documentation from the specific scientific context must be granted, an easily usable environment for LLM deployment, testing and tuning must be available, and methods for verification and optimization need to be deployed.

In KM3NeT, dedicated packages to address the first and last task are developed in the *InfoBasis* and *LLMTuner* packages, while the growing number of LLMs and their tools are used via the well-established combination of OpenWebUI and Ollama for interaction with and hosting of LLMs respectively, currently in an internal prototype setup. This is a change from the previously used AnythingLLM software (Schnabel 2025) due to the much larger flexibility in tools and LLM integration.

OpenWebUI[1] is a collaborative, self-hosted interface for LLMs that serves as a secure, private hub with multi-user management and a ChatGPT-like user interface. Its key feature is a modular pipelines architecture and an integrated RAG engine. It integrates with Ollama[2], which is a lightweight inference engine designed for the local execution of LLMs. It provides a stable OpenAI-compatible REST API, enabling local models to be served as scalable backend services for higher-level orchestration tools.

2.1. Data retrieval with *InfoBasis*

The InfoBasis package[3] is a Python-based framework designed for the systematic management of information retrieval from heterogeneous web sources and the local orga-

[1] https://github.com/open-webui/open-webui
[2] https://github.com/ollama/ollama
[3] https://git.km3net.de/jschnabel/InfoBasis

nization of data. Developed within the context of the KM3NeT neutrino observatory's Open Science System (Schnabel et al. 2021) and to retrieve information from internal ITC services, it serves as the foundational data management backbone for Retrieval-Augmented Generation (RAG) applications.

It contains and extensible interface framework in a modular "Talker" system based on an Interface class. Users can develop custom plugins to interact with various web servers, wikis, Git instances, or REST APIs. These Talkers encapsulate the logic for authentication, request handling, and data extraction.

The retrieved data can then be stored in a local data environment, including a configuration system for endpoint parameters and an SQLite database. The database serves as a central repository for metadata, indexing external resources, and tracking document versions.

Document processing and indexing is provided via an automated pipeline for document processing. A crosswalk mechanism facilitates the mapping of diverse API response structures into a standardized local database schema, ensuring consistency across different data sources.

2.2. LLM optimization with *LLMTuner*

The LLMTuner package[4] is another Python-based framework developed to extend and optimize Large Language Model (LLM) applications. Based on the InfoBasis package, it can facilitate the generation of vector databases by providing a link to retrieved data, and provide an interface to LLM implementations via API. Key components automate the creation of dedicated workspaces, management of vector database embeddings, and structured interaction with various LLM backends.

A dedicated Evaluator module, leveraging, as a starting point, the Hugging Face evaluate library, allows to benchmark model performance. It supports the creation of domain-specific test datasets to compare different prompt engineering strategies and LLM architectures.

2.3. Integration of the tools

The integration of these four tools creates a stack that integrates the access to scientific data with a tunable chat environment. InfoBasis as *Data Foundation handles* the "Talkers" that reach out to scientific repositories (GitLab, Wikis, REST APIs). The *Inference Engine* (Ollama) hosts the actual LLM and manages the computation. The *Optimization Layer* (LLMTuner) uses the inventory from InfoBasis to support the optimization of workspaces integrated in OpenWebUI. Finally, the *Gateway Layer* (OpenWebUI) provides the user interface and collaborative environment.

3. Example Applications of the Toolbox

3.1. Technological Level: Enhancing Data and Interface Accessibility

Utilizing the `infobasis` "Talker" architecture, the system can be interfaced with diverse scientific endpoints. This enables indexing of disparate sources and embedding in a vector database, facilitating semantic search capabilities that transcend traditional

[4] https://git.km3net.de/jschnabel/llmtuner/

keyword matching. Researchers can query the related information without being familiar with the exact terminology.

3.2. Ontological Level: Streamlining Methodology

The ontological level focuses on the meaning and methodology of scientific work, where the toolbox assists in the use of code and methods. Automated code review or assistance in writing can serve as a backbone for an analysis workflow assistant. It can perform preliminary code reviews of scientific software, checking for logical consistency or adherence to collaboration-specific coding standards. The toolbox also enables a RAG-based documentation assistant.

3.3. Contextual Level: Supporting Young Researchers and Outreach

At the contextual level, the toolbox addresses the socio-cultural and educational barriers to scientific entry. Here, the assistants can help democratizing science by supporting masterclasses and educational outreach as well as internal training. By providing multilingual support and the ability to generate simplified "abstractions" of complex results, the system helps non-experts—such as students or researchers from other disciplines—understand fundamental concepts in astroparticle physics.

4. Specializing for Neutrino Science

This toolbox is also contributed to the *PhysicsLLM* project[5] that aims to address the growing imbalance between rapid data acquisition in astro- and particle physics communities and the lagging implementation of FAIR principles. The project seeks to facilitate the transition from "Big Data" to "Smart Data" by developing an LLM-enhanced Research Data Management (RDM) toolkit. The use of the toolkit to generate a user-friendly analysis workflow discovering, enhancing and integrating data from the IceCube and KM3NeT neutrino observatory for the expert as well as the non-expert user will serve as a as guiding use case to develop LLM agents that can serve as facilitators in the access to scientific knowledge.

Note: This proceeding has been corrected and optimized using LLMs.

References

Schnabel, J. 2025, Large language models: New opportunities for access to science. eprint: 2501.07250, URL https://arxiv.org/abs/2501.07250

Schnabel, J., Gal, T., & Aly, Z. 2021, The km3net open science system. eprint: arXiv:2101.06751, URL https://arxiv.org/abs/2101.06751

[5] https://physics-llm.erumdatahub.de/

Astronomical Data Analysis Software and Systems XXXV
*ASP Conference Series, Vol. **VOL** Compiled: Tuesday 30th June, 2026@ 01:30*
K. Polsterer, D. Bohlender, S. Gwyn, A. Hincks and P. Teuben, eds.
*© **YEAR** Astronomical Society of the Pacific*

ZZZDEBUG!259.tex
Zacharias, M.|textbf
Montanari, A.
Wagner, S.J.
algorithm!machine learning
algorithm!machine learning!Generative AI
algorithm!machine learning!Large Language Model
databases!query language!PostgreSQL
computer languages!Python
web!development tools!PHP

pyhysics.tools – Enhancing the Findability of Astronomical Software

M. Zacharias,[1] A. Montanari,[1,2] and S.J. Wagner[1,3]

[1]*Landessternwarte, Zentrum für Astronomy Heidelberg, Universität Heidelberg, Königstuhl 12, 69117 Heidelberg, Germany;* *m.zacharias@lsw.uni-heidelberg.de*

[2]*Now at: Laboratoire Leprince-Ringuet, École Polytechnique, CNRS, Institut Polytechnique de Paris, F-91128 Palaiseau, France;* *montanari@llr.in2p3.fr*

[3]*Deutsches Zentrum für Astrophysik, Postplatz 1, 02826 Görlitz, Germany*

Abstract. FAIR software products are critical for the astronomical community in order to run all processes from data taking to analysis to simulations and interpretation. However, the Findability of software – especially within the vast realm of possible solutions – is the critical aspect, which is at best underdeveloped. Here, we introduce phyiscs.tools, a database of software products referenced in scientific publications. The database is filled with weblinks given in papers published on the arXiv, and additional information is pulled from the hosting server of the software product. The database is searchable in 3 ways: an SQL query, a search using an embedding model, and another one where the results from the embedding model are interpreted with a language model. The first option is very fast, but requires the knowledge of the name of the product. The second option allows for queries using key words, but can result in imprecise outputs. The third option refines the results of the second option and is therefore more precise, but takes longer. In this presentation, we will introduce the operational prototype which is running on a local CPU cluster. We will also highlight its capabilities and the ongoing development enabling fast and reliable Findability of software products used in the community.

1. Website structure

With the goal of building an efficient and lightweight website, which can be served by a SQL database and retrieve information via either a simple search or more advanced similarity pairing and Large Language Model (LLMs) reasoning, we designed the entire website structure in PHP and Python.

The index page is written in PHP, which dynamically generates the HTML structure and incorporates styling from a CSS stylesheet. The index shows the user the basic functionalities of `physics.tools`, which comprise four main actions: a standard query to the database, a similarity search, the combination of the latter with a re-ranking using a LLM local installation, and finally the possibility to request the addition of new tools to the database. The first functionality is simply served via a GET request; the other three work with POST requests pointing to different items. All the actions are executed via secondary PHP scripts, generating the HTML structure for the output while main-

taining the same style as the home page. The first functionality is executed directly as a query to PostgreSQL database. The similarity search, and its combination with the LLM re-ranking are considered as two distinct actions. For these two functionalities, as a simple query to the database is not sufficient, the execution of a Python pipeline is needed. This is triggered via the respective POST requests. More details about the search actions are given in Sec. 3. The fourth action simply prompts the user to fill out a form in order to submit their request.

As the website outputs a list of tools, the user can click on the desired one. For tools stored as GitHub repositories, we store information in our database directly using the GitHub API, as explained later in Sec. 2. For these database entries, the user can click on the tool name and be prompted to a secondary page with information on the tool, the articles citing the tool, and cross-citation information on more platforms. For tools that are not hosted on GitHub, it is rather difficult to define a structure for the extracted information. These tools are still stored in the database, as explained in Sec. 2, but the weblink corresponding to the tool citation in the article from which it was extracted is now referenced in the tool name. By clicking on it, the user will be prompted to the tool website.

2. Database

The database serving `physics.tools` is built from software citations harvested from articles published on arXiv, and extracted as weblinks. The articles are downloaded with a cron job running a Python pipeline. While it is difficult to select only weblinks pointing to software or software repositories (*i.e.* GitHub, Zenodo, or even personal websites hosting code), a first filter is applied to remove links that cite other articles. A second process of filtering is run by reorganising the links that are cited more than once as a single entry in the database. Once these steps are completed, a PostgreSQL database is compiled with several pieces of information extracted from the weblinks. If the cited software is hosted on GitHub, the Python GitHub API is used to retrieve the repository information. This is then processed to compile tables in the database specifically dedicated to the code hosted on GitHub. For other weblinks, the first few HTML tags on the website are extracted and combined in a single string. A local installation of the `deepseek-r1:14b` (Guo et al. 2025) large language model (LLM), running with `ollama` (Yang et al. 2025) on a server in Landessternwarte, is producing a summary, a list of keywords and other information to compile the tables dedicated to general software. `ollama` is a locally deployed AI model runner, which allows downloading and executing LLMs directly in local environments, even without the need for expensive hardware. In the future, APIs other than GitHub will be utilised to construct more specialised tables. These steps are routinely run on our machine to keep the database up-to-date: downloading new articles from arXiv, extracting and filtering weblinks, and compiling the database.

While the main information is entirely contained in the database, to improve the efficiency of the search engine, we build several `faiss` indexes (Douze et al. 2024). `faiss` is a library for efficient similarity search, built on algorithms searching through sets of vectors of any size. It can be run either on CPU or GPU memory. Each index is constructed as a `faiss.IndexHNSWFlat`, a flat index topped with a Hierarchical Navigable Small World (HNSW) structure to access elements more efficiently. The structure is defined according to the HNSW algorithm, a graph-based approximate nearest

neighbour search technique, which is later used to process the user's queries and search through the database. Each one of the indexes is paired with a `faiss.IndexIDMap`, to map the vectors in the index with the corresponding index in the original database. This is needed to pull the correct information from the database, following the user query. The `faiss` indexes accept embedded vectors. A word embedding is a representation of a word as a real-valued vector that encodes the meaning of the word in such a way that closer words in the vector space are expected to be similar in meaning. The embedding is necessary for the text values present in the SQL database, *i.e.* software descriptions and keywords. The word embedding for the indexes compilation is performed via a local installation of `nomic-embed-text` (Nussbaum et al. 2025), a large context length text encoder, running on the same `ollama` installation mentioned before.

3. Search engine

The search engine serving `physics.tools` works on three options: a standard query to the SQL database, a similarity search via the `faiss` indexes and the database, and the combination of the latter with a re-ranking using the local installation of the `gemma3` LLM running on `ollama`. Gemma is a lightweight, family of models from Google built on Gemini technology, built with a compact design to be deployed on resource-limited devices (Mesnard et al. 2024).

The first option is the fastest one, depending only on the response from the SQL database. This option searches through the tools' names in the database, prompting the output sorted by the position of the tool in the database. This option sends a GET request to the website index to query the database. The second option performs a similarity search between the user query and the `faiss` indexes, using a Python API that exploits the `nomic-embed-text` model. The embedded vectors in the indexes that are closest to the user query appear on top of the model output. This is then paired with the corresponding information in the database by extracting the numerical index value in the map. Although this option takes slightly longer than the first, it pairs user-provided keywords with the semantic meaning of the information in the `faiss` indexes. This option is executed via a POST request to the website index. The third option combines the similarity search of the second one with a re-ranking by the `gemma3` LLM. This further refines the output by adding a layer of reasoning typical of LLMs. Moreover, this simplifies the output, which can be useful for large databases. This option takes the longest among the three, and it could probably be executed faster with better hardware. Another alternative for a better level of reasoning is to include the `deepseek-r1:14b` model instead of `gemma3`. This, however, slows down the output for several minutes. This option is also executed via a POST request to the website index.

4. Conclusions

At the moment of the writing, the database powering `physics.tools` is being filled with more entries, to expand the reach of the search engine. Moreover, as the database grows, the different configurations for the search engine are tested to scale with the number of entries. We are also exploring new solutions to get the most efficient way of handling the two search engine options depending on the `nomic-embed-text` and `gemma3` models.

With exponential growth in the number of publications on the arXiv[1], the number of tools used by the scientific communities is also growing. While many of these are shared on public platforms such as GitHub, some researchers choose to host their publicly available software on university websites. The lack of a central platform where all the references to the software products used by the community can be explored feeds the process of "reinventing the wheel", when researchers write software from scratch for routines that were already satisfied by software developed by other teams, but not properly shared with the community.

The goal of `physics.tools` is to provide such a platform. As long as a tool is referenced in an arXiv publication, it should gain visibility in `physics.tools`, thereby enabling fast and reliable Findability of software products. Moreover, `physics.tools` implements statistics on the usage and cross-references of each tool in its database. With ongoing testing of the database and faiss index structures, and of the configuration and options for the search engines, `physics.tools` is poised to become a valuable contribution to the goal of creating a centralised open platform that collects references to the tools used by the scientific communities.

Acknowledgments. This work is funded in parts by the Deutsche Forschungsgemeinschaft (DFG, German Research Foundation), project number 460248186 (PUNCH4NFDI), and as part of Base4NFDI, grant number 521453681 (nfdi.software).

References

Douze, M., Guzhva, A., Deng, C., Johnson, J., Szilvasy, G., Mazaré, P.-E., Lomeli, M., Hosseini, L., & Jégou, H. 2024, ArXiv e-prints. eprint: 2401.08281

Guo, D., Yang, D., Zhang, H., Song, J., Zhang, R., et al. (DeepSeek-AI) 2025, Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning. eprint: 2501.12948, URL https://arxiv.org/abs/2501.12948

Mesnard, T., Hardin, C., Dadashi, R., Bhupatiraju, S., Pathak, S., et al. (Gemma) 2024, Gemma: Open models based on gemini research and technology. eprint: 2403.08295, URL https://arxiv.org/abs/2403.08295

Nussbaum, Z., Morris, J. X., Duderstadt, B., & Mulyar, A. 2025, Nomic embed: Training a reproducible long context text embedder. eprint: 2402.01613, URL https://arxiv.org/abs/2402.01613

Yang, M., J., M., Hiltgen, D., MacDonald, B., Gross, J., et al. 2025, ollama, https://github.com/ollama/ollama

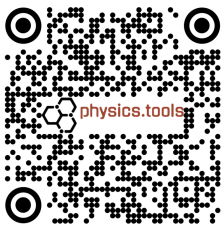


Figure 1. The physic.tools website is:https://hess.lsw.uni-heidelberg.de/physics.tools/

[1]https://arxiv.org/stats/monthly_submissions, accessed on 15 Jan 2026

Astronomical Data Analysis Software and Systems XXXV
*ASP Conference Series, Vol. **VOL** Compiled: Tuesday 30th June, 2026@ 01:30*
K. Polsterer, D. Bohlender, S. Gwyn, A. Hincks and P. Teuben, eds.
*© **YEAR** Astronomical Society of the Pacific*

ZZZDEBUG!261.tex
Saha, T.|textbf
Hessling, H.
conference!ADASS 2018
organisations!ASP

Software framework for detecting Pulsar Signals using Digital-Twin and Machine Learning

Tanumoy Saha[1] and Hermann Hessling [2]
[1]*HTW Berlin, Berlin, Germany*

[2]*DZA, Gorlitz, Lower Saxony, Germany*

Abstract. The astronomical data collected by large radio-telescopes can be complex and large in volume. To work with these data it is important to detect and identify the relevant information from it. This work aims to automatically identify pulsar signals using machine learning and simulation. The pipeline of our work starts with the example of single pulse detection from disperse graphs of pulsars. A software framework for digital twin of pulsar signals based on the spark model is presented, along with a pipeline based on UNet to segment and detect pulsar signals from the dispersion graphs. We plan to further develop the framework and make it significantly more efficient in order to identify more complex signal patterns.

1. Introduction

With the new telescopes like the Square Kilometer Array Observatory (SKAO), testing and discovering new celestial objects and rare astronomical signals becomes plausible. Recent advances in computation and machine learning aids further to that endeavor. In this project we have developed a simplistic Digital Twin based on well-established light house model and spark model of pulsars. A pipeline using legacy image-processing tools and CNN–based neural networks are also developed to detect real signals from real telescope data. The idea is to train the CNNs using synthetic data generated by the digital twin. This framework will allow scientists to add new models in the digital twin based on new theoretical model to check new hypothesis and also to detect new pulse features from data acquired by powerful telescopes like SKAO. Below we will briefly introduce the two important modules of ML-PPA, PulsarDT: the digtal twin and PulsarSA: the pipeline.

1.1. PulsarDT

Pulsar Digital Twin (or PulsarDT [1]) is a simple virtual magnetic non-relativistic rotator with emission regions at the magnetic poles as shown in Figure 1. The emission signals can be understood as rays emitted by a lighthouse Lorimer & Kramer (2004). The pattern of the emission beam is animated as drifting sparks inspired by the spark model of Gil & Sendyk (2000). The spark pattern and the drift velocity along other pulsar attributes like the pulsar orientation relative to earth axis and magnetic axis tilt

[1]PulsarDT Gitlab Repository https://gitlab.dzastro.de/punch/ml-ppa/pulsardt.

are the parameters of the Digital-Twin. The user can tune these parameters to engineer characteristic pulse patterns. The emission from the sparks are modeled as broadband spectrum following established power-law Lee et al. (2022), which travels through homogeneous interstellar medium (ISM). The radio emissions are then detected using the virtual telescope module which are represented as dispersion graphs.

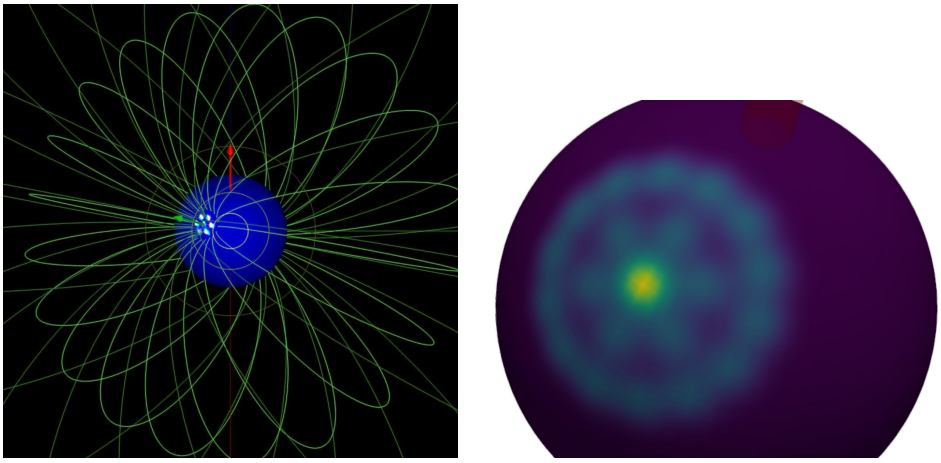


Figure 1. Digital Twin of Pulsar. *Left:* A snapshot of the virtual rotator with non-relativistic magnetic field lines. *Right:* A render of a spark pattern at the magnetic pole of the virtual rotator.

1.2. PulsarSA

Pulsar signal Segmentation and Analyze (or PulsarSA [2]) is a pipeline aimed at segmenting and identifying pulsar signals. The pipeline consists of three consecutive steps. The pipeline takes input of dispersion graph and identifies if and where the pulsar signal is present in the graph. The first step is to segment the dispersion graph using a CNN, for our purpose we choose a simple UNet. The second task is to filter out the segmented mask to remove noisy segments in the graph. The third step is to identify the type of the segments using connected component analysis

2. Current state

As of this paper publishing date, PulsarDT can be used to simulate drifting sub-pulse patterns and pulse-pattern engineering. Based on the spark pattern parameters chosen by the user, different pulse geometries can be synthesized. For example the user can arrange the sparks around the magnetic pole to generate different pulse geometries as shown in Figure 2. Similarly different drift velocities of the spark can be applied to the spark pattern to generate drifting sub-pulse pattern as shown in Figure 2. PulsarSA on the other hand comes with simple UNet type CNN for segmenting and filtering dispersion graphs. Legacy image-processing tool is used to further classify the segments. The steps are shown in Figure 3.

[2]PulsarSA Gitlab Repository https://gitlab.dzastro.de/punch/ml-ppa/pulsarsa

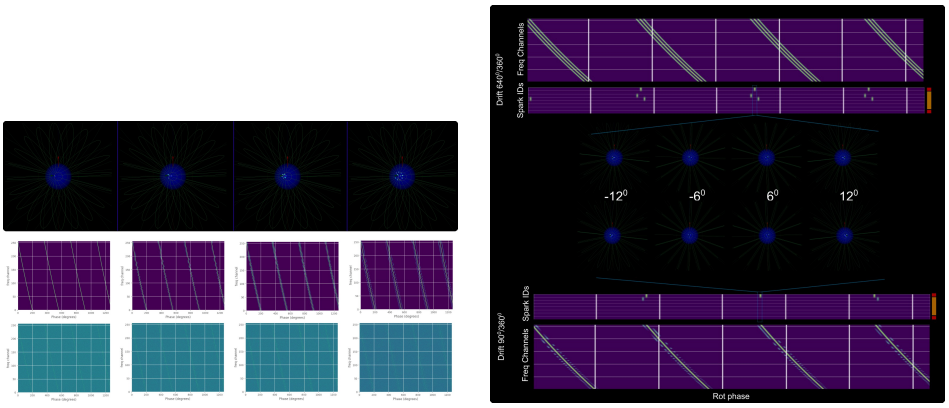


Figure 2. Current state of PulsarDT. *Left:* Pulse pattern modulation with different spark pattern geometries. (*Top:*) Snapshots of the virtual rotator with 1,3,7,13 spark patterns respectively from left to right. (*Middle:*) Dispersion graph for the respective spark patterns without noise. (*Bottom:*) Dispersion graph of the respective spark pattern with noise. *Right:* Pulse pattern modulation with different drift velocity. Two different scenarios are shown. (*Top Group:*) Drift velocity of 640°rotation around the magnetic axis of the drift patterns for 360°rotation of the pulsar around the rotation axis. This is denoted as 640°/360°drift velocity. The top plot shows the dispersion graph for 4 consecutive rotation phases. The plot in the middle shows the spark identified as spark IDs passing the view direction. The render plots in the bottom shows snaps at the consecutive rotation phases -12°, -6°, 6°, 12°. (*Bottom Group:*) Drift velocity of 90°/360°. The render plots in the top shows snaps at the consecutive rotation phases -12°, -6°, 6°, 12°. The plot in the middle shows the spark identified as spark IDs passing the view direction. The bottom plot shows the dispersion graph for 4 consecutive rotation phases -12°, -6°, 6°, 12°.

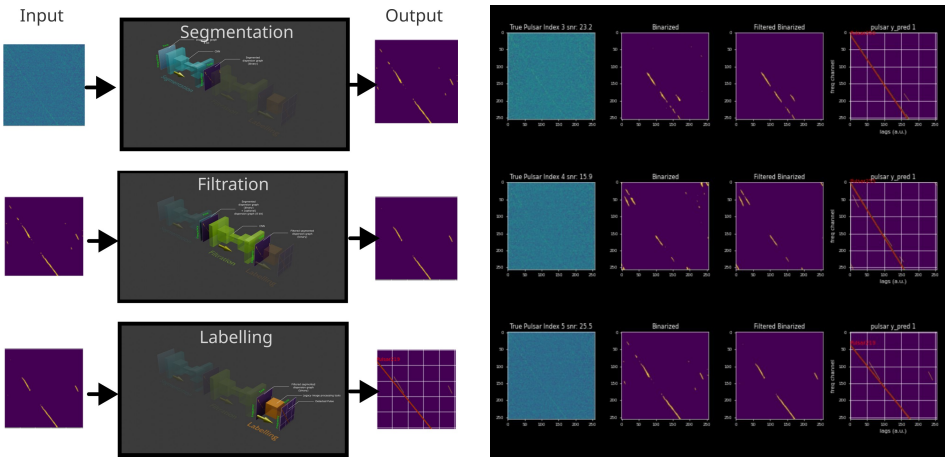


Figure 3. Current state of PulsarSA. *Left:* PulsarSA Pipeline serial steps: segmentation filtration and labeling. *Right:* Pipeline steps in real dispersion graphs.

3. Results and Outlook

To determine the efficiency of the pipeline we choose the example of single pulse signal from crab pulsar *B0531+21_58713_43190* acquired using the MeerKAT telescope. For this we generated synthetic datasets using PulsarDT with the same pulsar parameters and acquisition settings. To be specific we have generated ensemble of dispersion datasets with varying antenna sensitivity. Each ensemble contained ten-thousand dispersion graphs. Each ensemble sets were used then to train a simpler version of UNet for segmentation and filtering. The pipeline trained with synthetic data sets generated by PulsarDT showed F1-score > 0.85 for SNR>12 on real labeled datasets of crab pulsar shown in Figure 4. The current UNet has roughly 30,000 trainable parameters. The next steps include further optimizing the UNet to increase performance in wider DM ranges of pulsar signals while keeping a check in the number of trainable parameters. We are also working on a tuner module to incorporate multiple trained pipelines to be combined into one without the need of re-training.

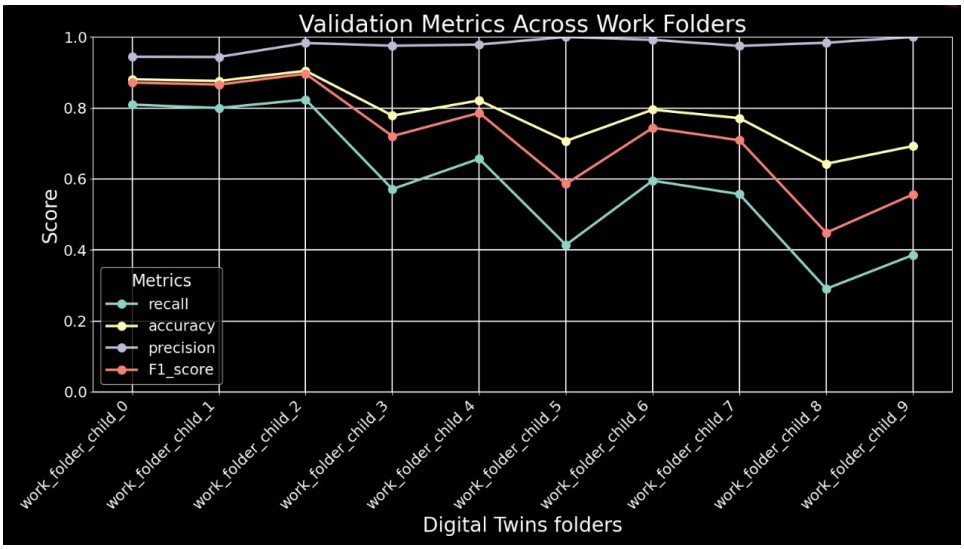


Figure 4. Performance of the software framework. The plot shows the performance metric scores recall, accuracy, precision and F1-score for the pipelines on real labeled Pulsar data with SNR>12 trained over different ensemble of synthetic datasets based on antenna sensitivity. The pipeline showed acceptable performance in ensemble dataset *work_folder_child_2* with F1-score > 0.85

References

Gil, J. A., & Sendyk, M. 2000, The Astrophysical Journal, 541, 351. URL https://doi.org/10.1086/309394

Lee, C. P., Bhat, N. D. R., Sokolowski, M., Swainston, N. A., Ung, D., Magro, A., & Chiello, R. 2022, Publications of the Astronomical Society of Australia, 39, e042

Lorimer, D. R., & Kramer, M. 2004, in Handbook of Pulsar Astronomy. URL https://api.semanticscholar.org/CorpusID:116894520

Astronomical Data Analysis Software and Systems XXXV
*ASP Conference Series, Vol. **VOL** Compiled: Tuesday 30th June, 2026@ 01:30*
K. Polsterer, D. Bohlender, S. Gwyn, A. Hincks and P. Teuben, eds.
*© **YEAR** Astronomical Society of the Pacific*

ZZZDEBUG!268.tex
Škoda, P.|textbf
Burakov, O.
Laiyk, A.
computing!architecture!RES
classification!spectral
algorithm!machine
learning!deep learning
archives!individual!LAMOST
software!containers

Small-Scale Science Platform for Active Deep Learning in Large Spectra Archives

Petr Škoda,[1] Oleksandr Burakov,[2] and Alisher Laiyk[2]

[1]*Astronomical Institute of the Czech Academy of Sciences, Ondřejov , Czechia; skoda@asu.cas.cz*

[2]*Faculty of Information Technology, Czech Technical University in Prague, Czechia*

Abstract. The emerging technology of Data Science platforms addresses the need to analyze Big Data directly where it is stored, because moving large volumes of data is nearly impossible. Typical platforms, like Pangeo, ESA DataLab, and SciServer are complex cloud-based systems, running on large clusters, providing hundreds of users access to petabyte-scale astronomical data archives. They use flexible orchestration of multiple containers to allow script-driven data processing and complex database queries, as well as interactive exploratory analysis via a web GUI.

Similar principles can be applied to smaller, task-specific infrastructures, particularly for machine learning experiments that demand continuous user interaction. A representative example is the active learning-based classification of millions of stellar spectra from large surveys. In such cases, iterative training requires annotators to label candidate spectra across multiple cycles while simultaneously accessing complementary data such as catalog metadata, images, and spectra from other surveys.

To address this need, we introduce the `ml-job-manager`, a dedicated multi-tier cloud platform designed for orchestrating parallel experiments with human-in-the-loop active deep learning on several million spectra from the LAMOST archives. The system adopts modern DevOps strategies, leveraging a microservices backend with RESTful asynchronous job control inspired by the IVOA UWS protocol. The entire environment in several containers can be rapidly deployed using Docker Compose, ensuring reproducibility and ease of installation.

1. Active Deep Learning

The Active Learning method is an approach to overcome the need for a large and representative training set, as is required by classical deep learning. It is based on idea that the algorithm will perform better if it is allowed to choose data for its training (Settles 2009).

A machine learning algorithm combined with active learning queries unlabelled data samples to be labelled by an *Oracle* (usually a human expert). The samples are selected in batches of a given size according to a certain informativeness measure. Commonly used is the *uncertainty sampling*, which selects data with the least certain labelling (based on information entropy)

A particular version of Active deep learning was introduced into astronomical spectra analysis and successfully applied to search for emission-line spectra in archive

of the LAMOST telescope by Škoda et al. (2020). The code used in their paper including the example Jupyter notebooks is available on GitHub (Podsztavek 2019).

2. Active Deep Learning Cycle

2.1. Labelling of New Training Samples by Oracle

The role of the Oracle (usually a human annotator) is to assign the correct label to the queried sample. The batch for Oracle labelling is selected from the sample where the network is least sure about the right classification. It is expected that the Oracle can make the decision immediately just by looking at the visual properties of the sample. In real cases, other data and metadata about the sample may be helpful to estimate the correct label. The batch samples from the labelled Oracle set are added to the training set and the network is retrained on them.

2.2. Performance Estimation Sample

In addition to the Oracle data set, there is another set of samples shown to to the expert as well. When it is correctly labelled, it is used for calculating the classification accuracy of the network. It helps to identify the moment when the training iterations should stop, as the performance of the algorithm is not further improving. The samples in this set are selected randomly, and after computing the performance their label is discarded.

2.3. Network Iterations

After the network is trained on the new training set with added samples from Oracle set, it is asked to provide classification of all data. They are ordered by the cross entropy, and a some small number (a batch) of samples from the top is converted into a an oracle set shown to the expert. This repeats until the accuracy no longer improves.

3. Cloud-based Science Platform `ml-job-manager`

A modern, cloud-based scientific platform `ml-job-manager` represents a comprehensive redevelopment of the now outdated system `VO-CLOUD` (Koza 2015, 2017) and its Active Deep Learning client (Mazel 2020).

This redevelopment was undertaken in two Bachelor's theses (Burakov 2025; Layik 2025) within the software and web engineering program at the Czech Technical University in Prague.

The current implementation features two main types of processing units. The primary-and most sophisticated-component is the Active Deep Learning worker, offering an interactive interface for labeling and visualization. The secondary unit, the Pre-processing worker, efficiently handles large-scale data tasks such as converting, rescaling, trimming, and padding millions of spectral samples.

The system shown on Fig. 1 is composed of a backend server and a web-based graphical frontend. Communication between these components utilizes a RESTful API, drawing inspiration from the IVOA Universal Worker Service (UWS) protocol (Harrison & Rixon 2016).

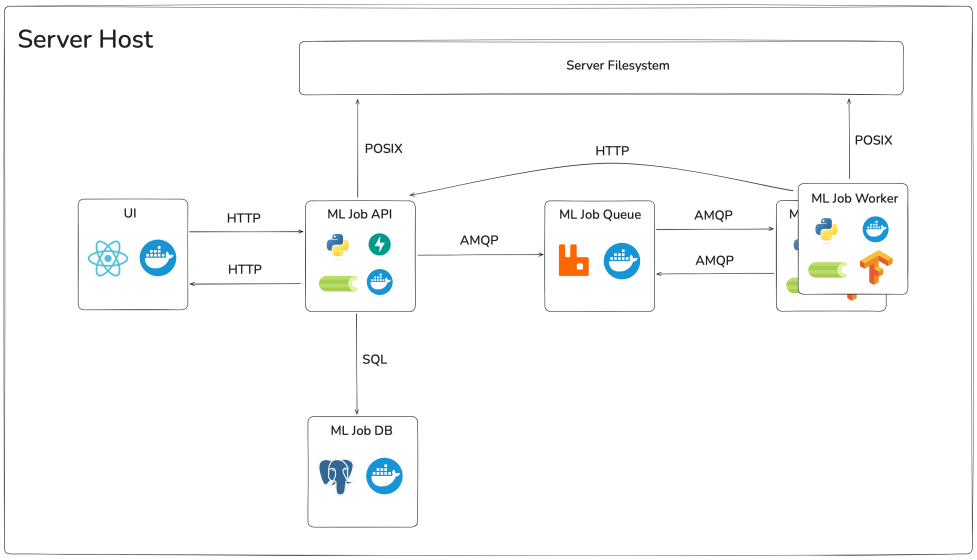


Figure 1. Schema of `ml-job-manager` science platform with logos of used frameworks

4. Technology Stack of `ml-job-manager`

The platform is based on microservice architecture combining a number of modern technologies, frameworks and libraries. Its development was following the best current DevOps practices. The whole system is easily deployed in several containers using `docker-compose` (or compatible `podman-compose`).

- **Implementation language:** Python 3.13
- **Package management:** Poetry
- **Relational database:** PostgreSQL 17
- **Object relational mapping and migrations:** SQLAlchemy, Alembic
- **Message broker:** RabbitMQ
- **Web framework:** FastAPI
- **Data validation & serialization:** Pydantic
- **Job orchestration:** Celery
- **Async I/O:** aiofiles
- **Configuration format:** JSON
- **Scientific data handling:** Astropy (FITS), h5py (HDF5), NumPy

- **Machine learning:** Scikit-learn, TensorFlow , Keras
- **Containerization:** Docker (or Podman) composer
- **Web Frontend:** React, TypeScript, Tailwind CSS, HTTPX, Plotly

5. Source Code

Source code of both backend and frontend modules are available on GitHub:

https://github.com/bursasha/ml-job-manager
https://github.com/sparki0/mj-job-manager-client-modules

6. Conclusions

The active deep learning of large spectral archives requires all the training iterations to be supported by the interactive labelling performed by the domain expert in role of the Oracle. The `ml-job-manager` cloud-based science platform provides the expert with the flexible and user-friendly visual environment allowing to run a number of experiments in parallel

Acknowledgments. Astronomical Institute of the Czech Academy of Sciences is supported by the project RVO 67985815. For testing the platform the spectra from the public LAMOST DR2 archive were used extensively.

References

Burakov, O. 2025, Science platform for machine learning of big astronomical data - cloud infrastructure. Bachelor's thesis. Czech Technical University in Prague, Faculty of Information Technology, URL https://doi.org/10.5281/zenodo.17420073

Harrison, P., & Rixon, G. 2016, IVOA universal worker service pattern. IVOA Recommendation, URL https://www.ivoa.net/documents/UWS

Koza, J. 2015, Design and implementation of a distributed platform for data mining of big astronomical spectra archives. Bachelor's thesis. Czech Technical University in Prague, Faculty of Information Technology, URL https://doi.org/10.5281/zenodo.17537102

— 2017, Interactive cloud-based platform for parallelized machine learning of astronomical big data. Master's thesis. Czech Technical University in Prague, Faculty of Information Technology, URL https://doi.org/10.5281/zenodo.17537158

Layik, A. 2025, Science platform for machine learning of big astronomical data - data analysis modules. Bachelor's thesis. Czech Technical University in Prague, Faculty of Information Technology, URL https://doi.org/10.5281/zenodo.17420100

Mazel, T. 2020, Cloud-based platform for active learning of astronomical spectra. Bachelor's thesis. Czech Technical University in Prague, Faculty of Information Technology, URL https://doi.org/10.5281/zenodo.17420002

Podsztavek, O. 2019, active-cnn source code for active deep learning on github.com, online. [Accessed 2025-11-05], URL https://github.com/podondra/active-cnn

Settles, B. 2009, Active Learning Literature Survey, Computer Sciences Technical Report 1648, University of Wisconsin–Madison

Škoda, P., Podsztavek, O., & Tvrdík, P. 2020, Astronomy & Astrophysics, 643, A122

Astronomical Data Analysis Software and Systems XXXV
*ASP Conference Series, Vol. **VOL** Compiled: Tuesday 30th June, 2026@ 01:30*
K. Polsterer, D. Bohlender, S. Gwyn, A. Hincks and P. Teuben, eds.
*© **YEAR** Astronomical Society of the Pacific*

ZZZDEBUG!374.tex
Enßlin, T.|textbf
Eberle, V.
Guardiani, M.
Westerkamp, M.
Bayesian
imaging
FOOBAR! ascl:1101.010

The Universal Bayesian Imaging Kit

Torsten Enßlin,[1–4] Vincent Eberle,[1,3] Matteo Guardiani,[1,3] and Margret Westerkamp[1,3]

[1]*Max Planck Institute for Astrophysics, Karl-Schwarzschild-Str. 1, 85748 Garching, Germany; ensslin/veberle/matteani/margret@mpa-garching.mpg.de*

[2]*Deutsches Zentrum für Astrophysik, Postplatz 1, 02826 Görlitz, Germany*

[3]*Ludwig-Maximilians-Universität München, Geschwister-Scholl-Platz 1, 80539 Munich, Germany*

[4]*Excellence Cluster ORIGINS, Boltzmannstr. 2, 85748 Garching, Germany*

Abstract. Bayesian imaging of astrophysical measurement data shares universal properties across the electromagnetic spectrum: it requires probabilistic descriptions of possible images and spectra, and instrument responses. To unify Bayesian imaging, we present the Universal Bayesian Imaging Kit (`UBIK`). Currently, `UBIK` images data from Chandra, eROSITA, JWST, and ALMA. `UBIK` is based on information field theory (IFT), the mathematical theory of field inference, and on `NIFTy`, a package for numerical IFT. `UBIK` provides sky models that are instrument independent and instrument interfaces that share common parts of their response representations. It is open source, can provide spatio-spectral image cubes, jointly analyses data from several instruments, and separates diffuse emission, point sources, and extended emission regions.

1. Universality of imaging

Currently, the different astronomical telescopes – be these in the radio, optical, X-ray, or any other wavelength regime – come with their specific imaging software. The task these systems address, however, is always the same, to convert the instrument data into an image of the sky while removing imperfection of the measurement process as far as possible. The data the telescopes deliver is never sufficient to specify the sky fully, there are always limitations due to various instrument effects. Thus, any generated image is a reconstruction of the celestial reality, partly uncertain itself and based on assumptions on the sky and the measurement process. A proper accounting of uncertainties is only possible in a strict probabilistic setting. This requires that the instrument and its statistical noise properties as well as the exploited prior knowledge are explicitly specified and used. Thus, imaging should be Bayesian. Since from an epistemological perspective no principal difference exists between the different imaging problems – in all data is converted into an image of the world using auxiliary knowledge – imaging should also be universal, meaning that the same methods are used across all measurement modalities.

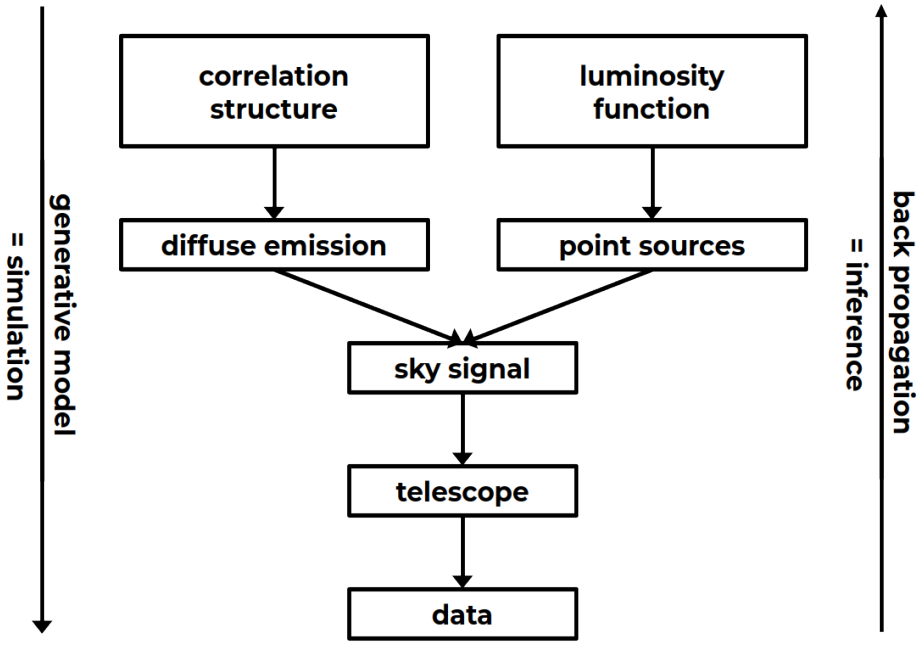


Figure 1. Graphical representation of a typical generative model used in astronomical imaging with UBIK.

2. Universal Bayesian Imaging Kit

For the aim of a unified approach to Bayesian imaging, we present UBIK, the **U**niversal **B**ayesian **I**maging **K**it (Eberle et al. 2024b), a Python-based software system (https://github.com/NIFTy-PPL/J-UBIK) that unifies imaging for a growing number of instruments and aims at enabling multi-instrument imaging and cross-calibration.

UBIK permits the user to describe the measurement situation in terms of a generative forward model, see Fig. 1. The model can be built out of a number of preexisting building blocks and user supplied elements. For example, the model shown in Fig. 1 might be used to image data from an X-ray telescope like Chandra. For this it consists of the following elements:

- A latent layer (not shown), which is the start of the forward model and is assumed to contain a priori independent standard Gaussian random variables, collectively described by a vector $\xi \in \mathbb{R}^n$, with $\mathcal{P}(\xi) = \mathcal{G}(\xi, \mathbb{1})$ that sources all variations of the model realizations.

- Transformations that shape these latent variables into model components $s = f(\xi)$ with the desired a priori statistics $\mathcal{P}(s) = \mathcal{P}(\xi) \left|\left|\frac{\partial f(\xi)}{\partial \xi}\right|\right|^{-1}_{\xi=f^{-1}(s)}$. These comprise the unknown spatial or spectral correlation function of the diffuse sky emission field, properties of the luminosity function of point sources, the diffuse and point source fields, calibration properties, and so on.

- A likelihood function $\mathcal{P}(d|s)$ that can be used to generate synthetic data d and that allows a comparison of expected detector imprint of an assumed sky signal with the observed data in light of the instrument noise statistics. This function needs to describe all relevant effects that imprint on the data, including eventually spatially varying point spread functions, in-homogeneous sky exposure, Poisson, Gaussian, or other noise processes, calibration fluctuations, cosmic ray hits, and more.

`UBIK` can run the model in forward direction and thereby generate synthetic skies and corresponding data sets. These can be used to demonstrate, verify, and characterize the imaging. The imaging itself is performed with real or simulated data entered at the end of the model. This data is compared to the expectation value of data for the model given the current state of the latent (and other) variable on the basis of the latter's log-likelihood. The discrepancy between real and expected data is back propagated through the generative process into the initial latent variable layer of the model. As this inversion is typically not unique – there exist a manifold of possible latent space variable configurations that are consistent with the data within the noise statistics – it is performed in a probabilistic fashion. This means that a sample of possible images are generated, drawn approximately from the Bayesian posterior distribution $\mathcal{P}(s|d) = \mathcal{P}(\xi|d)\,\|\frac{\partial f(\xi)}{\partial \xi}\|^{-1}_{\xi=f^{-1}(s)}$.

3. Variational inference

Since imaging involves the determination of many unknowns, the field values at all pixel locations as well as many parameters of the higher level of the generative model, the space of potential images is very high dimensional. In order to represent the posterior distribution over this space, `UBIK` usually resorts to variational inference (VI) as implemented in the `NIFTy` package.[1] There are two VI schemes available, *Metric Gaussian VI* (MGVI, Knollmüller & Enßlin 2019) and *geometric VI* (geoVI, Frank et al. 2021).

The former scheme, MGVI, approximates the posterior distribution by a Gaussian distribution $\mathcal{P}(\xi|\bar{\xi}) = \mathcal{G}(\xi - \bar{\xi}, \Xi(\bar{\xi}))$. The mean of the Gaussian, $\bar{\xi}$, contains the variational parameters that are optimized by minimizing the variational cross entropy $\mathcal{D}(\bar{\xi}) = \int \mathrm{d}\xi\, \mathcal{P}(\xi|\bar{\xi})\, \ln \frac{\mathcal{P}(\xi|\bar{\xi})}{\mathcal{P}(\xi|d)}$. The covariance $\Xi(\bar{\xi})$ of this Gaussian is approximated with the help of the Fisher information matrix $M_{\bar{\xi}} = \left\langle \frac{\partial \ln \mathcal{P}(\xi|d)}{\partial \xi} \frac{\partial \ln \mathcal{P}(\xi|d)}{\partial \xi}^{\dagger} \right\rangle_{(d|\bar{\xi})}$ as $\Xi(\bar{\xi}) = \left[\mathbb{1} + M_{\bar{\xi}}\right]^{-1}$. This form ensures that for directions in the latent space for which the posterior is Gaussian or not affected by the data, the covariance is exact. If this is true for all directions, the Gaussian posterior approximation is exact as well.

The latter scheme, geoVI, improves on this by constructing a further latent space that is reachable via an approximate normalizing transformation, $y = g_{\bar{\xi}}(\xi)$. This is done by constructing approximate Riemann normal coordinates according to the metric

[1] `NIFTy` stands for *Numerical Information Field Theory* (Selig et al. 2013; Arras et al. 2019; Edenhofer et al. 2024). It uses `JAX` for just in time compilation, GPU access, and automatic differentiation. *Information Field Theory* (IFT, Enßlin et al. 2009) is information theory for fields, the mathematical framework of Bayesian inference for field-like quantities.

$\mathbb{1} + M_\xi$ starting from an expansion point $\bar{\xi}$. The optimal expansion point is found by minimizing $\mathcal{D}(\bar{\xi})$ for the approximate distribution $\mathcal{P}(\xi|\bar{\xi}) = \mathcal{G}(g_{\bar{\xi}}(\xi), \mathbb{1}) \, \|\frac{\partial g_{\bar{\xi}}(\xi)}{\partial \xi}\|^{-1}$.

In both schemes, MGVI and geoVI, initially Gaussian samples are converted non-linearly into the model parameters, either via $s = f(\xi)$ in MGVI or with $s = f(g_{\bar{\xi}}^{-1}(y))$ in geoVI. These samples can then be used to calculate any kind of posterior summary statistics via $\langle F(s) \rangle_{(s|d)} \approx \frac{1}{N} \sum_{i=1}^{N} F(s_i)$, where $F(s)$ denotes the desired quantity and s_i any of the N samples.

4. `UBIK` components

`UBIK` provides model components to represent instrument response functions and their noise statistics and to describe various signals over one or higher dimensional domains like space, time, energy, and combinations thereof. It comes with ready to use instrument descriptions for a growing number of instruments, currently Chandra, eROSITA, JWST, and ALMA (Rüstig et al. 2024; Westerkamp et al. 2024; Eberle et al. 2024a; Guardiani et al. 2025).

Acknowledgments. M.G., V.E., and M.W. acknowledge financial support from the German Aerospace Center and Federal Ministry of Education and Research through the project Universal Bayesian Imaging Kit – Information Field Theory for Space Instrumentation (Förderkennzeichen 50OO2103). M.G. acknowledges support from the European Union-funded project mw-atlas under grant agreement No. 101166905. V.E. acknowledges funding through the German Federal Ministry of Education and Research for the project ErUM-IFT: Informationsfeldtheorie für Experimente an Großforschungsanlagen (Förderkennzeichen: 05D23EO1).

References

Arras, P., Baltac, M., Enßlin, T. A., Frank, P., Hutschenreuter, S., Knollmueller, J., Leike, R., Newrzella, M.-N., Platz, L., Reinecke, M., & Stadler, J. 2019, NIFTy5: Numerical Information Field Theory v5, Astrophysics Source Code Library, record ascl:1903.008

Eberle, V., Guardiani, M., Westerkamp, M., Frank, P., Freyberg, M., Salvato, M., & Enßlin, T. 2024a, arXiv e-prints, arXiv:2410.14599

Eberle, V., Guardiani, M., Westerkamp, M., Frank, P., Rüstig, J., Stadler, J., & Enßlin, T. A. 2024b, arXiv e-prints, arXiv:2409.10381

Edenhofer, G., Frank, P., Roth, J., Leike, R., Guerdi, M., Scheel-Platz, L., Guardiani, M., Eberle, V., Westerkamp, M., & Enßlin, T. 2024, The Journal of Open Source Software, 9, 6593

Enßlin, T. A., Frommert, M., & Kitaura, F. S. 2009, Phys.Rev.D, 80, 105005

Frank, P., Leike, R., & Enßlin, T. A. 2021, Entropy, 23, 853

Guardiani, M., Eberle, V., Westerkamp, M., Rüstig, J., Frank, P., & Enßlin, T. 2025, A&A, 703, A203

Knollmüller, J., & Enßlin, T. A. 2019, arXiv e-prints, arXiv:1901.11033

Rüstig, J., Guardiani, M., Roth, J., Frank, P., & Enßlin, T. 2024, A&A, 682, A146

Selig, M., Bell, M. R., Junklewitz, H., Oppermann, N., Reinecke, M., Greiner, M., Pachajoa, C., & Enßlin, T. A. 2013, A&A, 554, A26

Westerkamp, M., Eberle, V., Guardiani, M., Frank, P., Scheel-Platz, L., Arras, P., Knollmüller, J., Stadler, J., & Enßlin, T. 2024, A&A, 684, A155

Author Index

Škoda, P., **453**

Aboudan, A., 135
Abu, I., **189**
Accomazzi, A., 401
Adamo, A., 131
Adrian P. Schirra, A. S., **361**
Ahmed, S., 437
Ali, E., **151**
Ali, S., 85
Allen, A., 51
Allen, M., 293
Allen, T., 401
Allende-Prieto, C., 285
Alonso-Asensio, I., 285
Altieri, B., 267
Amati, L., 135
Andres, A., 175
Ao, Y., 325
Aranda, S.I., 151
Araya, M., 189
Arviset, C., 21, **175**, 267
ArÃ©valo, M., **247**, 297
Asad, M., 201
Ashkar, H., 217
Ates, O., 213, **213**
Avila, B. C., **221**

Bailer-Jones, C. A. L., 385
Baines, D., 21, 175, 297
Balawajder, T., 197
Barloy, N., **413**
Barreiro, M. F., 175
Basset, A., 357
Bassler, T., 59
Baumann, M., 51, 317, **397**
Becker, V.R., 151
Belanger, G., 193, 267
Bell, G. S., **259**
Bellm, E. C., 235
Bentley, M., 21, 175
Berger, J.-P., 417
Berriman, G. B., **143**
Berriman, G. B., 155
Berriman, G. B., 155
Berry, D. S., 259
Berry, K., 121
Bertocco, S., 293
Bhatawdekar, R., **21**, 175, 247
Bisero, S., 221
Blanco-Cuaresma, S., 401
Blay, P., 193
Boch, T., 51, **317**, 397
Bohlender, D., xiii
Boisson, C., 213
Booth, S. T., 121
Borciani, E., 189
Borra Sagar, B. S., **429**
Bot, C., 51
Bourgès, L., **417**
Broekgaarden, F., 321
Brownstein, J. B., 239
Bruijne, J., 21
Bruno, P., 189
Bryant, A., **47**, 335, 425
Buchbender, C., 343
Buchholz Elsa, B. E., 429
Buchholz, E., **409**
Buddelmeijer, H., 365
Bulgarelli, A., 135, 185, 189
Burakov, O., 453
Burns, E., 185
Busso, G., 437

Cánovas, H., 267
Cabello, C., 225
Calabretta, M. R., **347**
Calabretto, P., 135, 189
Calzolari, F., 135
Campos, H., 189

Canbolat Utku, C. U., 429
Canovas, J. L., 55
Caon, N., 285
Carasa, Ã., 247
Cardiel, N., 1, **5**, **225**
Carpenter, V. S., 201
Castaldini, L., **135**, 185, 189
Castro, N., 131
Castro, S., **111**
Cavadore, C., 433
Ceballos, M.T., 225
Ceballos-Merino, M. T., **1**
Cecconi, B., 293
Chang, C.-S., 121
Chelli, A., 417
Chen, S., 401
Cherinka, B., 239
Chiang, H-F., 235
Chillarón-Víctor, M., 5, 225
Chivvis, D., 401
Ciabattoni, A., 185
Ciric, A., 221
Cobo, B., 1
Coia, D., 267
Conforti, V., 189
Cook, J., 175
Cook, J. P., 247
Cook, K. W., 201
Corpora, M., 189
Costa, A., 189
Costa, M., 221
Coulson, J., **147**
Cristofari, P., 213
Cruzalèbes, P., 417
Cui, C., 171, 229
Curiban, S., 313
Currie, M. J., 259

Dauner, L., 209
Dauser, T., 209
de Angeli, F., 437
de Bony, M., 217
de Bruijne, J., 175, 267
de Marchi, G., 175
De Rosa, A., 135
de Santayana, R. P., **357**
de Teodoro, P., 55
de Vicente, A., 285
Demleitner, M., **9**, 293
Desing Raja Chithralekha, R. C., 429
de Macedo Alves, F., 401
Di Piano, A., 135, 189
Di Piano, A., 185
Docasal, R., 175
Doctor, M., 105
Donor, J., 239
Doser, B., **255**
Dreizler, S., 131
Drobek Mark, D. M., 429
Duarte, J., 321
Duvert, G., 417

Eberle, V., 457
Ehrlich, L., **105**
Emonts, B. H. C., **63**
Enßlin, T., **457**
Espinosa, J., 193, 247, 297
Evans, C., 297
Ezawa, H., 121

Falcón-Barroso, J., 285
Falco, R., 135, 185
Farago, A., 313
Fernandez, J., 55
Fernandez, V., 55
Fernique, P., **51**, 317, 381, 397
FernÃ¡ndez, M., 247, 297
FernÃ¡ndez, V., **297**
Fernández, M., 193
Ferruit, P., 267
Findeisen, K., **235**
Fiordoliva, F., 189
Fioretti, V., 135, 185, 189
Firusawa, J., 243
Ford, D., 437
Früh, J., 335
Friedl, P., 59
Früh, J., 47, 425
Furusawa, H., 243

Gülbahar, E. G., 209
Gómez, L., 121
Gómez, P., 267
Gaillard, N., **17**
Gallego, J., 5, 225
Gallego, S., 185
Gallozzi, S., 189
Garcia, J., 313

Gaudet, S., xiii
Geers, V. C., 121
Gelino, C. R., 155
Gelino, C. R., 155
Gesing, S., 321
Gianotti, F., 189
Gidofalvy, A., 59
Gil, P. P., 175
Gilles, G. L., 373
Giuliani, A., 189
Godard, J.-P., 433
Good, J. C., 155
Good, J. C., 155
Grant, C., 401
Graves, S. F., 259
Greiner, B., 335, **425**
Grezes, F., 401
Grillo, A., 189
Guardiani, M., 457
Gurgul, A., **389**
Gutiérrez-Sánchez, R., **285**
Gwyn, S., xiii
Gómez, P., **101**

Hütwohl, M., 335
Haberhauer, F., **365**
Hambly, N., 81
Han, J., **377**
Harnqvist, S., 81
Harrison, D., 437
Hartmann, D. H., 185
Haupt Lars, H. L., 429
Haupt, L., 409
Hayes, M., 131
Haynes, M. P., 139
He, B., 171
Heßling Hermann, H. H., 429
Heßling, H., 409
Heinl, H., 293
Henneken, E., 401
Hernandez, J. L., 17
Hessling, H., 449
Heßling, H., **251**
Hibbard, J. E., 121
Hilton, D. G., 201
Hincks, A., xiii
Hirzberger, J., 43
Hodgkin, S., 437
Holties, H. A., **205**
Hostetler, T., 401
Howard, E. L., 235
Hsieh, P.-Y., 313
Huerta, I., 189
Hunt, E. L., **201**
Hunter, T. R., 121
Huynh, S., **159**
Hütwohl, M., 425

Ibarra, A., 105
Ihnatiev, I., 121
Illingworth, M., **81**
Indebetouw, R., 121
Irwin, J., 437
Iserlohe, C., 47, 335
Iserlohe, I., 425
Ivan, I. B., **373**
Iyer, K., 39
Izumi, T., 121

Jacovich, T., 401
Jamal, S., **385**
Jaroschewski, I., 221
Jeanneau, A., 131
Jeschke, E., **289**
Jiang,H., **31**
Jiang, H., 39
Jochum, J., 201

Kackley, R., 289
Kagan, S., 139
Kajava, J., 193
Kasuga, T., 243
Katz, D. S., 321
Keida, A., 243
Keilig, T., 335, 425
Kelz, A., 131
Kemp, J., 39
Kent, B. R., **139**
Kepley, A. A., 121
Kern, J., 139
Kiendrébéogo, W., 221
KiendrÃ©bÃ©ogo, W., 217
Kierans, C. A., 185
Kim, M., 289
Kirsch, C., **209**
Kleinman, S.J., 151
Kleiver, L., 281
Knieling, B., 47, 335, 425

Koch, J., 401
Kocot, J., 197
Kotysz, K., 389
Krabbe, A., 335, 425
Kretlow, M., **393**, **421**
Kruk, S., 105, **267**
Kunneriath, D., 121
Kurtz, M., 401

López-Caniego, M., 267
Labiano, A., 297
Laiyk, A., 453
Lamiré, B., 213
Landais, G, 405
Landais, G., 381, 413
Le Montagner, R., 217
Leśniak, M., 197
Lemson, G., 277
Leon, I., 21, 175
Lepo, K., 201
Leschinski, K., 365
LeÃ³n, I., 247
Li, C., 13
Li, S., 229
Li, X., **305**
Lidman, C., 147
Lim, K-T., 235
Lin, Z., 325
Ling, C., 325
Lipnicky, A., 121, 313
Littenberg, T., 357
Liu, C., 325
Lockhart, K., 401
Lodi, M., 189
Loomis, R. A., 121
Lorenz, M., 209
Lucarelli, F., 189
Lutzgendorf, N., 357
Lynn, M., 155
LÃ³pez-Caniego, M., 297

Maggi, P., 281
Maksym, S., 267
Mandica, A., 189
Manning, A., 313
Marín-Galán, C., 285
Marchand, M., 381
Marchi, G., 21
Marinic, F., 267
Marinoni, A., 135
Martens, S., 131
Martinez-Castellanos, I., 185
Masson, A., 21, 175
Masters, J., 139
Matsson, P., 175
Mattioli, G., 189
Maud, L. T., 121
Mella, G., 417
Meneses, N. C., 59
Merín, B., 175
Merin, B., 21
MerÃn, B., 247, 297
Mi, L., 171
Michalik, H., 43
Michel, L., **281**
Mikołajczyk, J. P., 389
Milian, K., 197
Millenaar, T., 85
Miszalski, B., 147, 163
Mitschang, A., **277**
Modigliani, A., **339**
Molinaro, M., **293**
Montanari, A., 445
Moseley, R., **155**
Moseley, R., 155
Mourard, D., 417
Mowla, L., 39
Muders, D., **121**
Munari, K., 189
Muraleedharan, D., **43**
Murphy, D., 437
Mustafa, G., 135, 185

Nakazato, T., 121
Navarro, V., 175, 267
Neeleman, M., 121
Neights, E., 185
Nie, J., 325
Nieto, S., **55**
Niu, B., 325
Norman, H., 175

O'Brien, B., 81
O'Ryan, D., 101
O'Toole, S. J., 147
Oberto, A., **405**
Oliveira, J., 55
Olmedo, V., 55

Oluyide, T., 155
Oluyide, T., 155
Onken, C., 147
Ootsubo, T., 243
Ortiz, J.-L., 393
Osinde, J., 247

Pérez-Prieto, J. A., 285
Pal, K., 209
Palacin, H., 357
Panebianco, G., 135, 185
Paquin, J., **401**
Pareschi, G., 189
Parmiggiani, N., 135, **185**, 189
Pascual, S., 5, 225
Pastore, V., 189
Patel, P., 185
Peng, X., 325
Perez, H., 21, 175
Perez, P. A., 55
Pettorino, V., 267
Pineau, F., 51
Pineau, F. -X., **381**
Pizzo, R. P., 205
Plunkett, A., 313
Polsterer, K., xiii
Polsterer, K. L., 255, 263, 329
Prados, J., 55
Prieto, R., 55
PÃ©rez, H. H., 247

Rahman,M., 31
Rahman, M., **35**, 39
Raison, F., **59**
Reerink, J., 175, 267
Reinmann, N., 209
Remmelgas, K., A., 101
Ren, J., **325**
Richard, J., 131
Rios, C., 55
Rixon, G., 437
Rizzo, A., 135, 185
Ruhberg, L. E., 101
Ruiz, E., 55
Ruiz-Velasco, A. B. E., 217
Russo, F., 189

Saha, T., **449**
Saiz, J., 55
Savalle, R., 213, **433**
Savchenko, V., **301**
Saxton, R., 105
Sayes, N., 189
Schaaff, A., 293
Schindler, K., 335, 425
Schnabel, J., **441**
Schulz, B., **335**, 425
Schussler, F., **217**, 221
Schweder, F., 263, **329**
Schwinger, M., 59
Scora,J., 31
Scora, J., **39**
Scuderi, S., 189
Seglar-Arroyo, M., 217
Serene, A., 151
Servillat, M., 213
Shakunaga, T., 121
Shao, L., 325
Sharma, R., **369**
Sheckard, J., 139
Sheehan, P., 121
Sheng, K., 147
Shimada, K., 121
Shortridge, K., **93**, **97**
Siejkowski, H., **197**
Siles, A., 193
Smith, L. C., **437**
Solanki, S. K., 43
Solano, E., 285, 293
Song, Y.-S., 369
Sonnentrucker, P., 297
Spadoni, L., 135
Speck, D., 235
Spreeuw, H., **85**
Staub, J., 43
Stenborg, T. N., **73**
Stierhof, J. J. R., 209
Sto, J., 197
Stoehr, F., **263**, **313**
Stolyarov, V., 305
Sugimoto, K., 121
Sullivan, I. S., 235
Swain, M., 155

Tacchini, A., 189
Taghizadeh-Popp, M., **239**, 277
Takata, T., **243**
Tallon-Bosc, I., 417

Tan, S., 325
Tanaka, M., 289
Taniguchi, K., 121
Tao, Y., 171
Taylor, M. B., **309**
Templeton, M., 401
Teuben, P., xiii
Thakar, A, 277
Thakar, A., 239
Thalhammer, P., 209
Tocknell, J., 147, 159, **163**, **167**
Tomsick, J. A., 185
Torstensson, K., 121
Tosti, G., 189
Trujillo-Gomez, S., 255, 329

Usher, C., 89
Utrilla Molina, E., 81

Valls-Gabaud, D., 433
Van-Lane, P., **321**
Verbena, J. L., **343**
Vibhute, A., 121
Vidante, C., 131
Villa, F., **193**
Virgilli, E., 135

Wagg, T., 321
Wagner, S.J., 445
Walton, N., 437
Wambsganns, J., 293
Wang, R., 325
Wang, Y., **171**
Wasiak, F., 357
Wei, P., 325
Wei, S., 325
Weijmans, A., 239
Weilbacher, P. M., **131**
Wells, A., 111
Wenda, B., 197
Wendt, M., 131
Westendorp, C., 285
Westerkamp, M., 457
Wiedemann, M., 335, 425
Williams, P., 321
Williams, S. J., 121
Wilms, J., 209
Wipper, C., **89**
Withers,T., 31
Withers, T., 39
Wulf, E. A., 185
Wyrzykowski, Ł., 389

Xu, C., 325
Xue, R., 121

Yang, H., **229**
Yao, S., 325
Yoon, I., 121
Yoshino, A., 121

Zacharias, M., **445**
Zamorano, J., 225
Zaw, M. P, Z., 143
Zeile, O., 335, 425
Zhang, B., **77**
Zhang, J., 13
Zhang, Y., **13**
Zhao, B., 325
Zhou, Z., 325
Zieliński, P., 389
Zoglauer, A., 185
Zonca, A., 321
Zvirzdin, J., 201

Subject Index

AIPS, 347
AIPS convention, 347
AIPS++, 347
algorithm, 1
  automatic, 92
  CPCS, 389
  detection, 437
  k-d trees, 313
  machine learning, 445
    deep learning, 135, 186, 313, 329, 453
    Generative AI, 445
    Large Language Model, 445
    unsupervised, 313
  SExtractor, 389
  source finding, 85
applications
  Aladin Lite, 90, 239, 397
  ALMA Pipeline, 121
  CASA, 121
  DS9, 73
  GAIA, 259
archives
  design, 147
  individual
    ALMA Science Archive, 313
    GAIA, 389
    LAMOST, 453
    LOFAR, 205
    SOFIA, 335, 425
  Multiple
    Data Central Archives, 147
  multiple
    Infrared Science Archive (IRSA), 335
    NASA/IPAC Infrared Science Archive (IRSA), 425
    National Radio Astronomy Observatory (NRAO) data archive, 139
  reduced, 147
Artificial intelligence, 135
astronomy
  astrometry, 389
  exoplanet transits, 437
  exoplanets, 437
  gamma-ray, 135
  gamma-ray bursts, 186
  photometry, 389
  pulse detection, 1
  simulation, 329
  solar system, 393
  time domain, 281
  transients
    science alert, 85
  X-ray, 209, 281
Astrophysics Data System (ADS), 401

Bayesian, 457
BHTOM, 389
Big Data, 313

calibration, 325
CASA, 347
catalogues
  individual
    SDSS, 239
  services
    Gaia Archive Core Systems, 389
    VizieR, 90, 373
CFITSIO, 347
CHIPASS, 347
classification
  spectral, 453
Cloud Computing
  hybrid infrastructure
    ADASS 2025, 429

code
  repository
    Git, 425
    GitHub, 425
  reviews, 425
computer languages
  C++, 135, 209
  C++11, 135
  Java, 91
  JavaScript, 91, 239
  Julia, 421
  Python, 5, 93, 135, 159, 201, 225, 281, 335, 425, 445
  shell script, 73
computing
  architecture
    RESTful, 453
  cloud
    databases, 167
  cluster
    Docker Swarm, 189
  GPU, 437
  high-performance computing (HPC), 197
  resources, 277
conference
  ADASS 2018, 251, 449
cross-matching
  services, 239

data
  analysis, 1, 131
    real time (RTA), 135
  cube, 131
  Hierarchical Progressive Surveys (HiPS), 51, 397
  ingestion, 159
  management
    operations, 205
  pipelines, 135, 147
    processing, 85, 135, 335
    reduction, 335
    science, 335
  preservation, 335
  processing
    automated, 147, 235
  reduction, 131, 425
  simulated, 131
  tabular, 309
data analysis
  science platforms, 267
data formats, 309
  FITS, 73, 93, 413
    headers, 93
    keywords, 93
  HDF5, 135
  JSON, 91
databases
  Google BigQuery, 167
  individual
    Simbad, 90
    VizieR, 373
  MariaDB, 91
  Microsoft SQL Server, 167
  MongoDB, 189
  MySQL, 167
  Postgres, 159, 167, 335
    Citus, 167
    Greenplum, 167
  query language
    ADQL, 9
    PostgreSQL, 445
    SQL, 239
  SAP iQ, 167
  Sybase, 91
distortion, 347
Doxygen, 347

Earth
  atmosphere, 135
    absorption, 335
European Virtual Observatory, 335

FAIR, 293
FITS, 347
FITSWCS, 347
FOOBAR
  conference
    ADASS 2019, 21, 46, 63, 101, 105, 143, 171, 175, 193, 229, 243, 247, 305, 321, 357, 361, 369, 377, 417, 433
    ADASS 2025, 17, 339
  organizations
    ASP, 17, 21, 46, 63, 101, 105, 143, 171, 175, 193, 229, 243, 247, 305, 321, 357, 361, 369, 377, 417, 433

German Astrophysical Virtual Observatory (GAVO), 335

HEALPix, 347
High Performance Computing
  hybrid infrastructure
    ADASS 2025, 429
Hybrid infrastructure
  kubernetes and slurm
    ADASS 2025, 429

images
  processing
    photometry, 85, 389
    source extraction, 85, 335
imaging, 457
instruments
  individual
    WiFeS, 147
    X-IFU, 1
international collaboration, 151
International Virtual Observatory Alliance (IVOA), 51, 293

journals, 35

KM3NeT, 441

libraries
  Angular, 91
  Astropy, 413, 425
  HDF5, 135
  HEALPix, 397
  numpy, 425
  OpenGL, 97
  scipy, 425
  Vulkan, 97
  WebGL, 397
LLM, 441

methods
  Gaussian distributions, 329
  indexing
    Hierarchical Triangular Mesh (HTM), 167
    PgSphere, 167
    Q3C, 167

observatories
  ground-based
    ALMA, 121
    E-ELT, 365
    East Asian Observatory, 259
    JCMT, 259
    LOFAR, 205
    Rubin, 235
    Rubin Observatory, 151
    SKA, 397
    SOFIA, 335, 425
    Southern African Large Telescope (SALT), 163
  operations, 163
  space-based, 325
    AGILE, 135
    ATHENA, 1
    Gaia, 437
    JWST, 39
    PLATO, 437
observing
  scheduling
    queue, 89
organizations
  ASP, 251, 449
  EASST, 389
  ESA, 437
  NASA, 425

packages
  Docker, 135, 189
  Megaradrp, 5
  ORAC-DR, 259
  pandas, 425
  Starlink, 259
  Teareduce, 225
pipelines, 135, 281
  Rubin Prompt Processing, 235
portals
  CDS, 90
projects
  ASTRI, 189
  Astro-CC, 293
  EOSC, 293
  ESCAPE, 293
  SciServer Compute, 239
protocols
  HTTP, 91
  ObsCore, 413

satellites, 135

Sloan Digital Sky Survey (SDSS), 239
software, 111
  containers, 453
  Data reduction, 225
  design, 151, 209
    user requirements, 153
  development, 209
    DevOps, 163, 281
  framework
    rtapipe, 135
  frameworks
    Spark, 167
  image processing, 85, 313, 389
  Linux, 73
  logging, 151
  macOS, 73
  observation planning, 365
  open source, 197, 277
  operations, 163
  packages
    DaCHS, 9
  performance, 159
  project tracking
    JIRA, 163
  simulation, 209
  support, 163
  tools
    CCDPhot, 389
    Django, 159
    OpenGL, 97
    PyQt, 425
    Qt, 97
    Vulkan, 97
  user interfaces, 153
    tooltips, 153
    web-based, 197, 277
  workflows, 197
stars
  light curves, 437
surveys, 325
  Euclid, 267
  LSST, 151, 235
  Sloan Digital Sky Survey (SDSS), 239

target of opportunity, 92
techniques
  API, 91
  dashboards, 153
  data challenges, 153
  event detection, 1
tensorflow, 135

Virtual Observatory (VO), 293, 335
  standards, 9
visualization, 93
  3D, 97
  web-based, 239

WCSLIB, 347
web
  Apache NiFi, 147
  application, 151, 197, 239, 397
  browser, 91
  development tools
    Google Web Toolkit (GWT), 197
    PHP, 445
  social media, 201
  technologies, 239

ZZZDEBUG
  003.tex, 1
  004.tex, 5
  007.tex, 9
  009.tex, 13
  010.tex, 17
  019.tex, 21
  020.tex, 31
  021.tex, 35
  022.tex, 39
  025.tex, 43
  030.tex, 47
  039.tex, 51
  040.tex, 55
  041.tex, 59
  043.tex, 63
  045.tex, 73
  047.tex, 77
  048.tex, 81
  053.tex, 85
  055.tex, 89
  057.tex, 93
  058.tex, 97
  060.tex, 101
  062.tex, 105
  064.tex, 111
  066.tex, 121

067.tex, 131
071.tex, 135
072.tex, 139
073.tex, 143
076.tex, 147
079.tex, 151
081.tex, 155
082.tex, 159
084.tex, 163
085.tex, 167
086.tex, 171
088.tex, 175
090.tex, 185
094.tex, 189
095.tex, 193
096.tex, 197
097.tex, 201
102.tex, 205
107.tex, 209
109.tex, 213
117.tex, 217
123.tex, 221
128.tex, 225
129.tex, 229
130.tex, 235
133.tex, 239
149.tex, 243
151.tex, 247
155.tex, 251
157.tex, 255
160.tex, 259
162.tex, 263
168.tex, 267
171.tex, 277
176.tex, 281
177.tex, 285
184.tex, 289
186.tex, 293
187.tex, 297
190.tex, 301
191.tex, 305
196.tex, 309
198.tex, 313
204.tex, 317
205.tex, 321
207.tex, 325
208.tex, 329
209.tex, 335
210.tex, 339
215.tex, 343
216.tex, 347
217.tex, 357
219.tex, 361
229.tex, 365
231.tex, 369
232.tex, 373
233.tex, 377
234.tex, 381
236.tex, 385
237.tex, 389
239.tex, 393
240.tex, 397
241.tex, 401
243.tex, 405
244.tex, 409
245.tex, 413
246.tex, 417
247.tex, 421
248.tex, 425
254.tex, 429
255.tex, 433
256.tex, 437
257.tex, 441
259.tex, 445
261.tex, 449
268.tex, 453
374.tex, 457

ASCL Index

Aladin
ascl:1112.019, 51, 397
AMPEL
ascl:2005.015, 389
Astropy
ascl:1304.002, 373, 413
Astropy: ascl:1304.002, 425
ATLAS
ascl:1911.013, 389

CARTA
ascl:2103.031, 51, 313
CASA
ascl:1107.013, 111, 121, 347
CFITSIO
ascl:1010.001, 347
CUBE
ascl:1805.018, 51

DaCHS
ascl:1804.005, 9, 335
DS9
ascl:0003.002, 51, 73

exoplanet
ascl:1910.005, 437

FOOBAR
ascl:1101.010, 17, 21, 46, 63, 93, 97, 101, 105, 143, 171, 175, 193, 197, 201, 229, 243, 247, 281, 285, 305, 325, 329, 339, 357, 361, 369, 377, 381, 393, 401, 417, 421, 433, 457

GAIA
ascl:1403.024, 259, 389, 437
GALAXY
ascl:1904.002, 39
GAMMA
ascl:2104.024, 186
GLASS
ascl:1806.009, 39

HEALPix
ascl:1107.018, 51, 186, 397

IRAF
ascl:9911.002, 73

MEGAlib
ascl:1906.018, 186
Montage
ascl:1010.036, 155

ORAC-DR
ascl:1310.001, 259

pile-up
ascl:1806.014, 135
PLATO
ascl:1506.003, 437
PORTAL
ascl:2003.006, 51
PySE
ascl:1805.026, 85
PyWiFeS
ascl:1402.034, 147

Q3C
ascl:1905.008, 167

rate
ascl:1904.014, 135

Saada
ascl:1111.003, 413
SAOImage DS9: ascl:0003.002, 425
SIRENA
ascl:2307.013, 1
SIXTE
ascl:1903.002, 209

Spectra
ascl:1701.003, 397
Starlink
ascl:1110.012, 259
SWIFT
ascl:1805.020, 389

TOPCAT
ascl:1101.010, 373
Transit
ascl:1704.008, 437

WCSLIB
ascl:1108.003, 347

ASTRONOMICAL SOCIETY OF THE PACIFIC



The Astronomical Society of the Pacific is an international, nonprofit, scientific, and educational organization. Some 130 years ago, on a chilly February evening in San Francisco, astronomers from Lick Observatory and members of the Pacific Coast Amateur Photographic Association—fresh from viewing the New Year's Day total solar eclipse of 1889 a little to the north of the city—met to share pictures and experiences. Edward Holden, Lick's first director, complimented the amateurs on their service to science and proposed to continue the good fellowship through the founding of a Society "to advance the Science of Astronomy, and to diffuse information concerning it." The Astronomical Society of the Pacific (ASP) was born.

The ASP's purpose is to increase the understanding and appreciation of astronomy by engaging scientists, educators, enthusiasts, and the public to advance science and science literacy. The ASP has become the largest general astronomy society in the world, with members from over 70 nations.

The ASP's professional astronomer members are a key component of the Society. Their desire to share with the public the rich rewards of their work permits the ASP to act as a bridge, explaining the mysteries of the universe. For these members, the ASP publishes the Publications of the Astronomical Society of the Pacific (PASP), a well-respected monthly scientific journal. In 1988, Dr. Harold McNamara, the PASP editor at the time, founded the ASP Conference Series at Brigham Young University. The ASP Conference Series shares recent developments in astronomy and astrophysics with the professional astronomy community.

To learn how to join the ASP or to make a donation, please visit http://www.astrosociety.org.

ASTRONOMICAL SOCIETY OF THE PACIFIC

MONOGRAPH SERIES

Published by the Astronomical Society of the Pacific

The ASP Monograph series was established in 1995 to publish select reference titles.
For electronic versions of ASP Monographs, please see
http://www.aspmonographs.org.

INFRARED ATLAS OF THE ARCTURUS SPECTRUM, 0.9-5.3μm
eds. Kenneth Hinkle, Lloyd Wallace, and William Livingston (1995)

VISIBLE AND NEAR INFRARED ATLAS
OF THE ARCTURUS SPECTRUM 3727-9300Å
eds. Kenneth Hinkle, Lloyd Wallace, Jeff Valenti, and Dianne Harmer (2000)

ULTRAVIOLET ATLAS OF THE ARCTURUS SPECTRUM 1150-3800Å
eds. Kenneth Hinkle, Lloyd Wallace, Jeff Valenti, and Thomas Ayres (2005)

HANDBOOK OF STAR FORMING REGIONS: VOLUMES I and II
THE NORTHERN and SOUTHERN SKIES
ed. Bo Reipurth (2008)

TWENTY YEARS OF ADASS
ed. Ian Evans (2013)

SCIENCE WITH A NEXT-GENERATION VERY LARGE ARRAY
ed. Eric J. Murphy and the ngVLA Science Council (2018)

A complete list and electronic versions of ASPCS volumes may be found at
http://www.aspbooks.org.

All book orders or inquiries concerning the ASP Conference Series, ASP Monographs, or International Astronomical Union Volumes published by the ASP should be directed to:

Astronomical Society of the Pacific
390 Ashton Avenue
San Francisco, CA 94112-1722 USA
Phone: 800-335-2624 (within the USA)
Phone: 415-337-2126
Fax: 415-337-5205
Email: service@astrosociety.org

For a complete list of ASP publications, please visit http://www.astrosociety.org.