The Journal of Open Source Software

pyDARTdiags: A Python package for manipulating observation sequences and calculating observation-space diagnostics for the Data Assimilation Research Testbed (DART)

Helen Kershaw [1¶], **Marlee Smith** [1], **Isaac Arseneau** [2], **and Lukas Kugler** [3]

1 NSF National Center for Atmospheric Research, Boulder CO, United States of America **2** Texas Tech University, United States of America **3** University of Vienna, Austria ¶ Corresponding author

DOI: 10.21105/joss.09439

Software

- Review
- Repository
- Archive

Editor: Marcel Stimberg

Reviewers:

- @peanutfun
- @timothyas

Submitted: 05 August 2025
Published: 19 February 2026

Summary

pyDARTdiags is a Python package for manipulating observation sequences and calculating observation-space diagnostics for the Data Assimilation Research Testbed (DART) (J. Anderson et al., 2009; Gharamti et al., 2025; UCAR/NSF NCAR/CISL/DAReS, 2025).

Data assimilation is a scientific technique that combines observations (such as measurements from weather satellites, buoys, radar, or other sensors) with predictions from numerical models to produce an improved estimate of the state of a system (J. L. Anderson, 2009; Evensen, 2003). It is widely used in fields like meteorology, oceanography, hydrology, and environmental science.

During assimilation, the model state is transformed into observation space by interpolating the model state to each observation location. Observation space diagnostics are used to visualize observations and model predictions (in observation space), and compare their statistical properties both before and after assimilation. Thereby, these diagnostics are key tools to model evaluation and prediction tasks.

The Data Assimilation Research Testbed (DART) is a widely used community software facility for ensemble data assimilation. pyDARTdiags is a Python package for manipulating observation sequences and calculating observation-space diagnostics for DART. It provides tools to read, manipulate,and analyze DART observation sequence files using familiar, modern Python libraries. With pyDARTdiags, users can extract data, compute diagnostics, and create visualizations, all within reproducible Python workflows that integrate seamlessly with the broader scientific ecosystem.

Statement of need

DART merges diverse and complex observations into an internal data format called “observation sequence” files, which inherit metadata from each observation. Observation sequence files including the model states are also generated, wherein the model states have been transformed into the observation space for direct comparisons and analysis. While DART provides robust tools for data assimilation, its observation sequence files are not easily accessible for users wishing to perform custom diagnostics or integrate with Python-based workflows.

DART’s observation sequence files are central to its workflow, but their format and complexity can make them challenging to manipulate and analyze outside of the DART ecosystem. Existing DART tools for calculation of observation-space diagnostics are written in Fortran with visualization in MATLAB, which may not be freely accessible to all users. To manipulate and analyze the data using open-source tools a converter/interface is required. pyDARTdiags is a Python package created to address this need by providing a single package to process and visualize observation space diagnostics.

PyDARTdiags ingests observation sequences into an ObsSequence object which contains the metadata about an observation sequence and a pandas DataFrame (The pandas development team, 2025) containing all the data for the observations.

This provides several advantages over the existing Fortran+MATLAB DART software:

- Providing Python routines for reading and writing DART observation sequence files allows for the manipulation of observation sequences interactively via DataFrames using popular, open-source data science libraries.
- Synthesizing the manipulation, analysis, and visualization of observation sequence files into a single Python workflow improves portability and flexibility over the fractured Fortran/MATLAB workflow.
- Enabling calculation of observation-space statistics (e.g., RMSE, bias, total spread) on a DataFrame enables Data Assimilation researchers and users to write custom diagnostics based on DataFrames. By decoupling the observation sequence file format from the DataFrame-based analysis, pyDARTdiags ensures that updates to the DART file format do not disrupt user-created diagnostic routines.
- Supporting both static (via Matplotlib, Hunter, 2007; McKinney, 2010) and interactive (via Plotly, Plotly Technologies Inc., 2015) plotting, facilitates the processing of observational datasets, quality control, and gaining insights about the spatial distribution of outliers or other (technical) anomalies.
- Facilitating reproducible, scriptable workflows for observation-space diagnostics enables the inclusion in Jupyter notebook workflows. A concrete use case is its integration into the the CESM Regional Ocean and Carbon Configurator with Data Assimilation and Embedding (CROCODILE, CROCODILE-CESM, 2025) project, which is a community platform for accelerating observationally-constrained regional ocean modeling. pyDARTdiags is used for observation space diagnostics and model-to-observation comparison in the Jupyter notebook workflows for this project.

Examples for manipulating observation sequences, visualizing observational data, and generating diagnostic plots can be found in the pyDARTdiags examples gallery. A detailed description of the available diagnostic statistics and available plotting functions is provided in the user guide. The pyDARTdiags source code is available at https://github.com/NCAR/pyDARTdiags.

Acknowledgements

We thank the DART team, Enrico Milanese (Woods Hole Oceanographic Institution), and the broader data assimilation community for their feedback and contributions.

This material is based upon work supported by the U.S. National Science Foundation under Grant No. 2311382, “Collaborative Research: Frameworks: A community platform for accelerating observationally-constrained regional oceanographic modeling”, and by the NSF National Center for Atmospheric Research, which is a major facility sponsored by the U.S. National Science Foundation under Cooperative Agreement No. 1852977.

References

Anderson, J. L. (2009). Ensemble Kalman filters for large geophysical applications. *IEEE Control Syst. Mag.*, *29*, 66–82. https://doi.org/10.1109/MCS.2009.932222

Anderson, J., Hoar, T., Raeder, K., Liu, H., Collins, N., Torn, R., & Avellano, A. (2009). The Data Assimilation Research Testbed: A community facility. *Bulletin of the American Meteorological Society*, *90*(9), 1283–1296. https://doi.org/10.1175/2009BAMS2618.1

CROCODILE-CESM. (2025). *CROCODILE: Regional ocean and carbon cycle modeling and data assimilation within the community earth system model*. GitHub repository. https://github.com/CROCODILE-CESM

Evensen, G. (2003). The ensemble Kalman filter: Theoretical formulation and practical implementation. *Ocean Dyn.*, *53*, 343–367. https://doi.org/10.1007/s10236-003-0036-9

Gharamti, M. E., Kershaw, H., Raeder, K., Raczka, B., Johnson, B., Smith, M., Anderson, J., Amrhein, D., Collins, N., Hoar, T., Gaubert, B., Grooms, I., & Kugler, L. (2025). The Data Assimilation Research Testbed: A robust, scalable software facility with groundbreaking capabilities for model-data integration. *Bulletin of the American Meteorological Society*, *106*(11), E2328–E2345. https://doi.org/10.1175/BAMS-D-24-0214.1

Hunter, J. D. (2007). Matplotlib: A 2D graphics environment. *Computing in Science & Engineering*, *9*(3), 90–95. https://doi.org/10.1109/MCSE.2007.55

McKinney, W. (2010). Data structures for statistical computing in Python. In S. van der Walt & J. Millman (Eds.), *Proceedings of the 9th Python in Science Conference (SciPy 2010)* (pp. 56–61). https://doi.org/10.25080/Majora-92bf1922-00a

Plotly Technologies Inc. (2015). *Collaborative data science*. Plotly Technologies Inc. https://plot.ly

The pandas development team. (2025). *pandas-dev/pandas: Pandas*. https://doi.org/10.5281/zenodo.3509134

UCAR/NSF NCAR/CISL/DAReS. (2025). *The Data Assimilation Research Testbed* (Version v11.11.2). https://doi.org/10.5065/D6WQ0202