








HuracanPy: A Python package for reading and analysing cyclone tracks

Stella Bourdin ^{1*} and Leo Saffin ^{2*}

¹ University of Oxford, Department of Physics, Oxford, UK  ² University of Reading, Department of Meteorology, Reading, UK  ¶ Corresponding author * These authors contributed equally.

DOI: [10.21105/joss.08174](https://doi.org/10.21105/joss.08174)

Software

- [Review](#) 
- [Repository](#) 
- [Archive](#) 

Editor: [Hugo Ledoux](#) 

Reviewers:

- [@nmstreethran](#)
- [@malmans2](#)

Submitted: 09 January 2025

Published: 24 June 2025

License

Authors of papers retain copyright and release the work under a Creative Commons Attribution 4.0 International License ([CC BY 4.0](#)).

Summary

HuracanPy is a Python module for working with cyclone track data, where cyclone tracks are recorded as a list of points in space and time. HuracanPy provides a powerful loading function that supports all the most commonly used track data formats the authors are aware of, and loads the tracks as Xarray datasets ([Hoyer & Hamman, 2017](#)). As such, we draw from the power of Xarray's built-in functions and the functionality of other libraries that work with Xarray, such as MetPy ([May et al., 2022](#)). We also provide functions specific to cyclone analysis, which are also gathered as methods in an Xarray accessor. These include functions to subset or interpolate track data, to add useful information, to compute common diagnoses, and to make quick exploration plots. With this package, we hope to make track data analysis more accessible and more reproducible.

Defining track data

Cyclones can be defined as pointwise features in space and time. Therefore, their trajectory can be recorded as a list of points in space and time. We call “track data” a list of such points. In many cases, the list not only consists of position in space and time, but also contains additional information about the cyclone, such as intensity. When the dataset contains data about several cyclones, each individual track is identified by a “track ID”.

Statement of need

In meteorological and climatological research about cyclones of any kind, we have been using different file formats and conventions for track data. HuracanPy's first purpose is to be able to read all these files in a common Python framework.

There is also a lack of a common analysis tool fostering reproducible analysis of such track data. For this reason, HuracanPy's second purpose is to offer a suite of functions to build analysis workflows.

As such, HuracanPy is mainly an analysis tool that comes after the tracking and up to preliminary plots in a research project. Our aim is to make track data analysis more accessible, and to promote good reproducibility practices.

Comparison with other packages

HuracanPy aims at becoming a standard community tool covering track data reading and analysis, which did not exist before. Prior to developing HuracanPy, we were not aware of any standard way of working with track data in python. Even within our own project and

research groups we found that most people had different ways of loading and working with track data. A few packages covered part of HuracanPy's functionality, with less flexibility in terms of supported track formats and analysis functions:

- Tropycal ([Tropycal, n.d.](#)) is a Python package designed for analysing and visualising tropical cyclone tracks from observations and operational forecasts. However, it does not have the flexibility to read data from other sources. For example, the loading of track data is part of the initialisation of the custom class used by Tropycal. We see HuracanPy as complementary to Tropycal and a future plan is to support conversion of track data to Tropycal objects with HuracanPy.
- The Cyclone Metrics Package (CyMeP) ([Zarzycki et al., 2021](#)) is a software suite providing a command-line function to provide standard assessment graphs. It can only run with CSV track data, and while it provides a good tool for rapid comparison of datasets, it is not flexible enough for exploratory and in-depth scientific analysis that HuracanPy wants to support. An assessment package based on merging functionalities from CyMeP and HuracanPy is in development.

Description

HuracanPy is built on Xarray. It loads track data as an Xarray dataset, and provides an Xarray accessor to provide cyclone-specific analysis methods. This allows us to draw from Xarray's performance and existing methods, and also means that we can handle track attributes which are not just scalar.

Loading data

HuracanPy currently supports loading tracks from:

- CSV (comma-separated values) files where each row corresponds to a cyclone point and each column to a feature.
- NetCDF (Network Common Data Form) files following CF (Climate and Forecast) conventions ([Hassell et al., 2017](#)), specifically the formats for trajectory data described in H.4 ([CF Conventions, n.d.](#)).
- NetCDF files similar to the CF conventions can also be loaded, provided they use specific naming for the track ID.
- Text files from TempestExtreme's StitchNodes (GFDL format, see ([Ullrich et al., 2021](#)))
- Text files from TRACK ([Hodges, 1995, 1999](#))
- Text files from IRIS ([Sparks & Toumi, 2024](#))
- Text files using the "original HURDAT" format [here](#) and used in particular within the European Centre for Medium-range Weather Forecast (ECMWF).

HuracanPy can also load IBTrACS ([Gahtan et al., 2024](#)) from an online source, or from an embedded file.

Tracks are loaded as an Xarray dataset with one main dimension called record.

HuracanPy can also save the track data as CSV and NetCDF files.

Analysis

HuracanPy contains various modules for analysis and visualisation of track data. * Manipulating data (`huracanpy`). Loading the track data as an Xarray object means the user can easily use Xarray's methods such as `.where` to subset. However, we found it useful to add a specific function that allows for subsetting based on a specific track ID, and on some track properties. We also added a specific function to interpolate the track to a different time resolution. * Adding track info (`huracanpy.info`). HuracanPy includes functions to add

common geographical and categorical information to the tracks, such as whether a point is over land or the ocean, or which category it belongs to in a defined convention. * Computing common diagnostics (`huracanpy.calc`). HuracanPy includes functions to compute track duration, genesis or lifetime maximum intensity attributes, translation speed, lifecycles and rates. * Tropical-cyclone specific diagnostics (`huracanpy.tc`). These include accumulated cyclone energy ([ACE](#)), pressure-derived ACE (PACE) ([Zarzycki et al., 2021](#)), [Saffir-Simpson categories](#), and Klotzbach pressure categories ([Klotzbach et al., 2020](#)). * Comparing two (or more) sets of tracks (`huracanpy.assess`). HuracanPy provides a one-to-one track matching function. From such a matching table, scores of Probability of Detection (also called Hit Rate) and False Alarm Rate can be computed. We also provide a function for computing the delay in onset and offset between the matched tracks. * Plot (`huracanpy.plot`). HuracanPy currently embeds a small number of functions, which are meant mostly for preliminary visualisation. These include plotting the track points and the track density. Due to the package being built on Xarray, Xarray's plot function can be used. We leave it to the user to build more elaborate plots using their preferred libraries, noting that Xarray in particular works well with seaborn.

Where possible, functions have been made unit aware by using the accessor and wrapper from MetPy ([May et al., 2022](#)), which converts the inputs into a Pint quantity ([Pint, n.d.](#)) internally using the units from the Xarray variable's attributes.

Most of HuracanPy's analysis functions can be called through an Xarray accessor named `hrcn`. The accessor provides a simpler syntax because it knows the variable types and default naming conventions, whereas the functions are generally more flexible with types and naming.

Perspectives

We welcome suggestions of other track data formats to support, and are happy to receive contributions of code to do so.

At the moment, HuracanPy only includes tropical-cyclones specific diagnostics, but we are more than happy to receive contributions and feedback from other communities, including but not restricted to extratropical cyclones, polar lows, medicanes, subtropical cyclones, etc. While the motivation for HuracanPy is analysis of cyclone tracks, the data format and workflow should also apply to other types of trajectories, such as Lagrangian air-mass trajectories, and we would be interested in supporting common data formats for these. Main avenues for a potential v2 are supporting multi-dimensional data (e.g. snapshots from TempestExtremes' NodeFileCompose ([Ullrich et al., 2021](#)), WiTRACK footprints ([Befort et al., 2020](#))).

Acknowledgements

Both authors acknowledge financial support from the Hurricane Risk Amplification and Changing North Atlantic Natural disasters (Huracán) NERC-NSF large grant n°NE/W009587/1 (NERC) & AGS-2244917 (NSF). The package is named after this project. This work also benefited from the TROPICANA program, supported by the Institut Pascal at Université Paris-Saclay, under "Investissements d'avenir" ANR-11-IDEX-0003- 01.

The package includes code that was originally developed in the scope of SB's thesis, which was funded by the Commissariat à l'Energie Atomique et aux Energies Alternatives (CEA), and the EUR IPSL-Climate Graduate School through the ICOCYCLONES2 project, managed by the ANR under the "Investissements d'avenir" programme with the reference ANR-11-IDEX-0004 - 17-EURE-0006.

References

- Befort, D. J., Kruschke, T., & Leckebusch, G. C. (2020). Objective identification of potentially damaging tropical cyclones over the western north pacific. *Environmental Research Communications*, 2(3), 031005. <https://doi.org/10.1088/2515-7620/ab7b35>
- CF Conventions. (n.d.). Retrieved December 2, 2024, from <http://cfconventions.org/Data/cf-conventions/cf-conventions-1.11/cf-conventions.html>
- Gahtan, J., Knapp, K., Schreck, C., Diamond, H., Kossin, J., & Kruk, M. (2024). *International best track archive for climate stewardship (IBTrACS) project, version 4r01*, NOAA national centers for environmental information.
- Hassell, D., Gregory, J., Blower, J., Lawrence, B. N., & Taylor, K. E. (2017). A data model of the Climate and Forecast metadata conventions (CF-1.6) with a software implementation (cf-python v2.1). *Geosci. Model Dev.*, 10(12), 4619–4646. <https://doi.org/10.5194/gmd-10-4619-2017>
- Hodges, K. I. (1995). Feature Tracking on the Unit Sphere. *Monthly Weather Review*, 123(12), 3458–3465. [https://doi.org/10.1175/1520-0493\(1995\)123%3C3458:FTOTUS%3E2.0.CO;2](https://doi.org/10.1175/1520-0493(1995)123%3C3458:FTOTUS%3E2.0.CO;2)
- Hodges, K. I. (1999). Adaptive Constraints for Feature Tracking. *Monthly Weather Review*, 127(6), 1362–1373. [https://doi.org/10.1175/1520-0493\(1999\)127%3C1362:ACFFT%3E2.0.CO;2](https://doi.org/10.1175/1520-0493(1999)127%3C1362:ACFFT%3E2.0.CO;2)
- Hoyer, S., & Hamman, J. (2017). Xarray: N-D labeled Arrays and Datasets in Python. *Journal of Open Research Software*. <https://doi.org/10.5334/jors.148>
- Klotzbach, P. J., Bell, M. M., Bowen, S. G., Gibney, E. J., Knapp, K. R., & Schreck, C. J. (2020). Surface pressure a more skillful predictor of normalized hurricane damage than maximum sustained wind. *Bulletin of the American Meteorological Society*, 101(6), E830–E846. <https://doi.org/10.1175/bams-d-19-0062.1>
- May, R. M., Goebbert, K. H., Thielen, J. E., Leeman, J. R., Camron, M. D., Bruick, Z., Bruning, E. C., Manser, R. P., Arms, S. C., & Marsh, P. T. (2022). MetPy: A meteorological Python library for data analysis and visualization. *Bulletin of the American Meteorological Society*, 103(10), E2273–E2284. <https://doi.org/10.1175/BAMS-D-21-0125.1>
- Pint: Makes units easy. (n.d.). Retrieved January 7, 2025, from <https://pint.readthedocs.io/en/stable/>
- Sparks, N., & Toumi, R. (2024). The ImpeRIal college storm model (IRIS) dataset. *Scientific Data*, 11(1), 424. <https://doi.org/10.1038/s41597-024-03250-y>
- Tropycal. (n.d.). Retrieved June 12, 2025, from <https://tropycal.github.io/tropycal/>
- Ullrich, P. A., Zarzycki, C. M., McClenny, E. E., Pinheiro, M. C., Stansfield, A. M., & Reed, K. A. (2021). TempestExtremes v2. 1: A community framework for feature detection, tracking, and analysis in large datasets. *Geoscientific Model Development*, 14(8), 5023–5048. <https://doi.org/10.5194/gmd-14-5023-2021>
- Zarzycki, C. M., Ullrich, P. A., & Reed, K. A. (2021). Metrics for Evaluating Tropical Cyclones in Climate Data. *Journal of Applied Meteorology and Climatology*, 60(5), 643–660. <https://doi.org/10.1175/JAMC-D-20-0149.1>