The Journal of Open Source Software

yieldplotlib: A unified library for exoplanet yield code visualizations

Corey Spohn [1*], **Sarah Steiger** [2*], **and Alex R. Howe** [1,3,4]

1 Goddard Space Flight Center, United States **2** Space Telescope Science Institute, United States **3** The Catholic University of America, United States **4** Center for Research and Exploration in Space Science and Technology, NASA/GSFC, United States ***** These authors contributed equally.

DOI: 10.21105/joss.09401

Software

- Review
- Repository
- Archive

Editor: Warrick Ball

Reviewers:

- @ben-cassese
- @gbergsten

Submitted: 01 April 2025
Published: 11 December 2025

Summary

NASA's next flagship observatory, the Habitable Worlds Observatory (HWO), aims to detect and characterize ~25 habitable zone planets. The total number of habitable-zone planets detected is referred to as the exo-Earth "yield" and accurate yield estimates will be critically important to the mission's success. Tools like the Altruistic Yield Optimizer (AYO) and EXOSIMS provide these yield estimates but differ in language, methods, and outputs. `yieldplotlib` provides a unified library that can visualize the inputs and outputs of these yield codes in a complete, descriptive, and accessible way.

Statement of need

To evaluate different designs for HWO (Feinberg et al., 2024), yield codes such as AYO (Stark et al., 2014) and EXOSIMS (Delacroix et al., 2016) calculate the expected exo-Earth yield for each architecture. While these yield codes have the same goal, their implementations are so different that validation has become a major obstacle for the community.

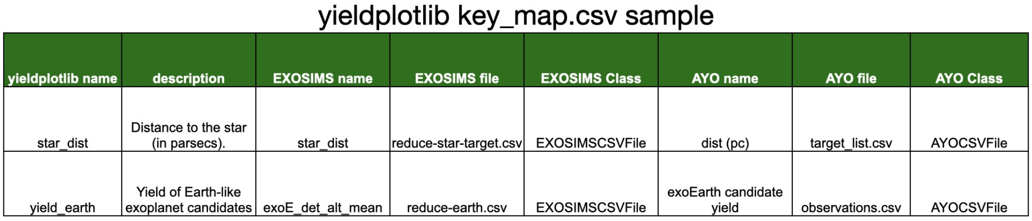
Previous cross-calibration efforts have already provided valuable insights. Stark et al. (2025) compared the internal exposure time calculations of AYO and EXOSIMS and revealed previously unknown discrepancies. `yieldplotlib` is a more ambitious continuation of that work. It is a Python library capable of easily accessing hundreds of important quantities from both AYO and EXOSIMS.

To visualize the inputs and outputs of AYO and EXOSIMS, `yieldplotlib` uses a custom loading and parsing structure to easily access equivalent data across both codes. This allows `yieldplotlib` to communicate the results of yield codes to the broader community and produce publication-quality plots without manually processing the complex underlying data. Currently `yieldplotlib` contains modules for analyzing AYO and EXOSIMS, but can be easily extended to support other yield codes.

Methods and functionality

Parsing and getting values

`yieldplotlib` provides a loading system with a unified interface for accessing data from the yield codes. The system internally manages the complex and inconsistent file structures of the AYO and EXOSIMS inputs and outputs so that users can access data with simple and consistent queries. For collaboration purposes, the valid queries are managed in a Google Sheet (see Figure 1) and automatically processed into a universal key map.



yieldplotlib key_map.csv sample

yieldplotlib name	description	EXOSIMS name	EXOSIMS file	EXOSIMS Class	AYO name	AYO file	AYO Class
star_dist	Distance to the star (in parsecs).	star_dist	reduce-star-target.csv	EXOSIMSCSVFile	dist (pc)	target_list.csv	AYOCSVFile
yield_earth	Yield of Earth-like exoplanet candidates	exoE_det_alt_mean	reduce-earth.csv	EXOSIMSCSVFile	exoEarth candidate yield	observations.csv	AYOCSVFile

Figure 1: Example portion of the `yieldplotlib` key map containing the mappings between AYO, EXOSIMS, and `yieldplotlib` parameters.

Plotting

Generic and comparison plots

`yieldplotlib` extends the widely used Python plotting library Matplotlib, leveraging its extensive customization options and the familiarity many users will already have with it. The `yieldplotlib` generic plots (`ypl_plot`, `ypl_scatter`, and `ypl_hist`) are used for single yield run visualizations.

To compare multiple yield runs, `yieldplotlib` also provides a set of flexible comparison plots that create multi-panel figures to quickly identify discrepancies.

In order to generate summary plots quickly, `yieldplotlib` provides a command-line interface and plotting pipeline to create a suite of commonly used yield plots.

Plotting scripts

`yieldplotlib` contains scripts for generating common plots used in yield code visualizations. These allow users to instantly compare AYO and EXOSIMS results as motivated by the rapid progress of HWO's ongoing architecture trade studies. These scripts also function as examples for users who want to adapt the generic `yieldplotlib` methods to generate bespoke visualizations.

Figure 2 compares the mission's "habitable-zone completeness", the probability that the simulated mission's observations would detect a planet in the habitable zone if one exists, as calculated by the two yield codes side-by-side. Figure 3 shows histograms of the total number of detected planets found as a function of planet type for the two codes though, in this example, the inputs to each code also differ resulting in the seen discrepancies.

Yield code inputs have a profound impact on calculated yield and plotting them is important to ensure consistency. Yield input packages (YIPs), a set of files that describe coronagraph performance, can also be loaded and accessed in `yieldplotlib` to compare how different codes process the same input coronagraph. Figure 4 shows a comparison of the calculated coronagraph throughput for both the two codes studied, and as accessed via an additional YIP analysis tool called `yippy`. Smaller throughputs result in less planet light on the detector which can result in lower yields.

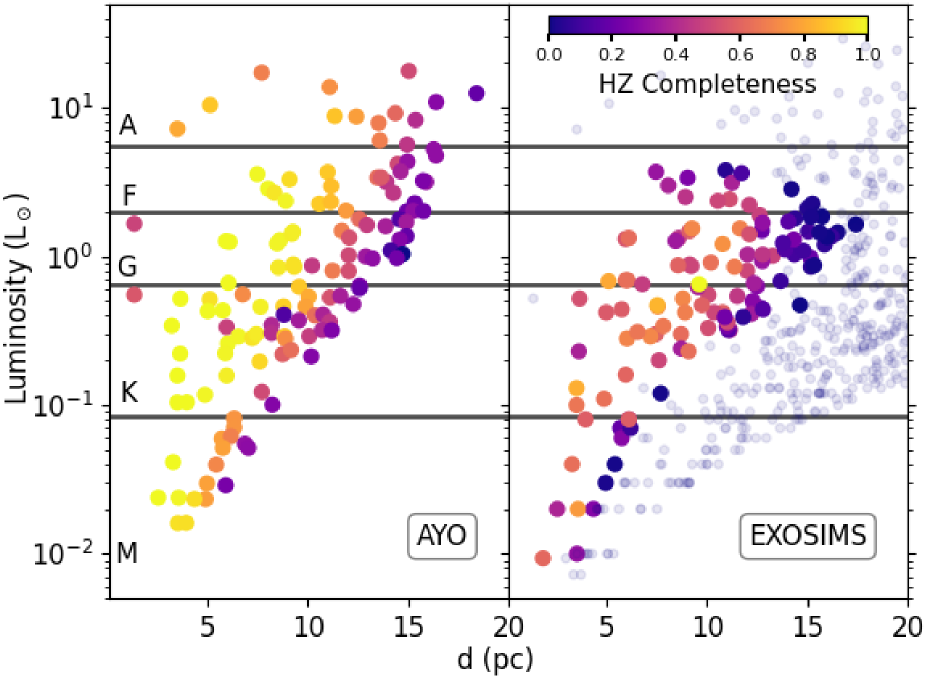


Figure 2: Plot of the Habitable-Zone (HZ) completeness as a function of host star luminosity (in units of Solar luminosity) and distance (in parsecs). Here the AYO results are on the left and the EXOSIMS results are on the right.

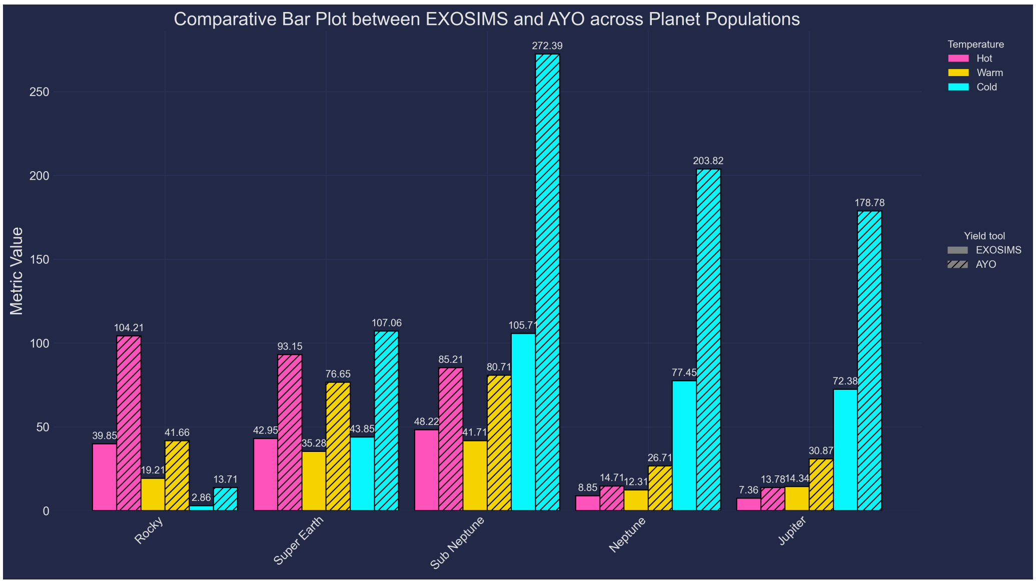


Figure 3: Bar chart comparing AYO and EXOSIMS planet yields for different classes of planets. This plot demonstrates `yieldplotlib`'s usage of the `mplcyberpunk` color scheme as an alternative dark-mode style. Note that this is a demonstration plot only, the AYO and EXOSIMS inputs shown are not directly comparable.

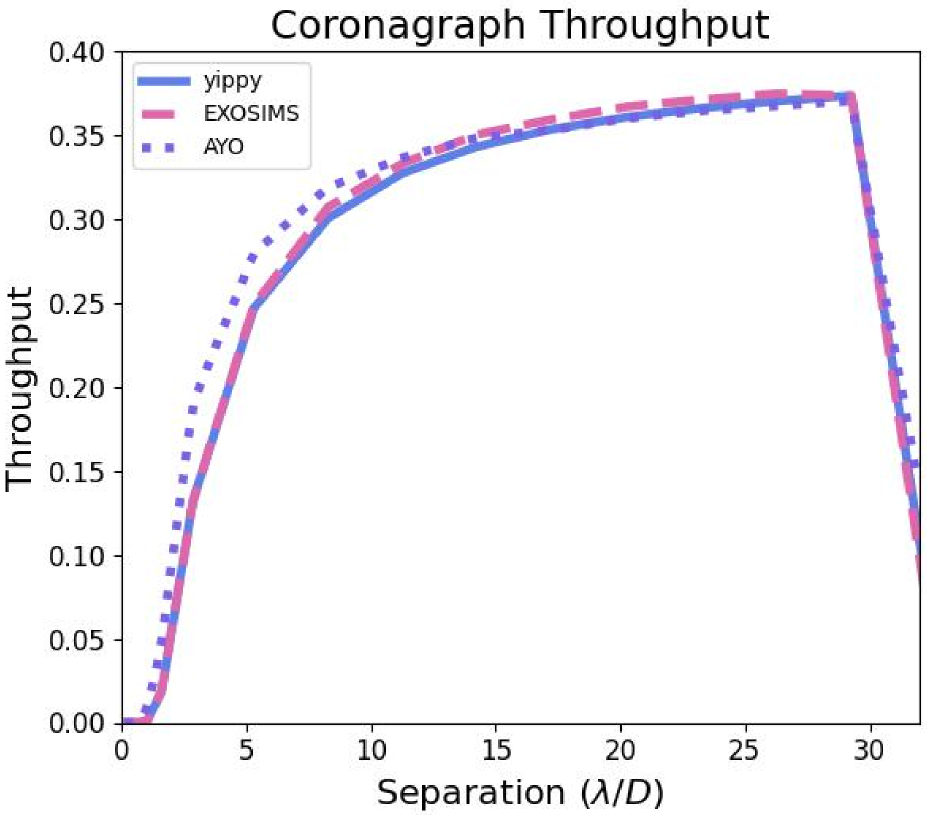


Figure 4: Core throughput vs. separation (in λ/D) for the same coronagraph when processed by AYO, EXOSIMS, and calculated by a tool named `yippy`. The differences are due to the different interpolation methods used and the definition of aperture over which the throughput is calculated. This highlights the types of insights that tools like `yieldplotlib` can help to uncover.

Future work

A new cross-calibration study of yield codes is being organized and will utilize `yieldplotlib`. This study will be vital to the ongoing HWO architecture trade studies and ensure more reliable and robust yield estimates.

Acknowledgements

Corey Spohn's research was supported by an appointment to the NASA Postdoctoral Program at the NASA Goddard Space Flight Center, administered by Oak Ridge Associated Universities under contract with NASA. Sarah Steiger acknowledges support from an STScI Postdoctoral Fellowship.

The authors would also like to acknowledge Christopher Stark, Dmitry Savransky, Rhonda Morgan, and Armen Tokadjian for providing consultation on the AYO and EXOSIMS repositories. They would also like to thank Justin Hom, and the rest of the Exoplanet Science Yields Working Group (ESYWG) for their valuable feedback and discussions as well as Susan Redmond for the development of the sample Yield Input Package.

References

Delacroix, C., Savransky, D., Garrett, D., Lowrance, P., & Morgan, R. (2016). Science yield modeling with the Exoplanet Open-Source Imaging Mission Simulator (EXOSIMS). In G. Z. Angeli & P. Dierickx (Eds.), *Modeling, systems engineering, and project management for astronomy VI* (Vol. 9911, p. 991119). https://doi.org/10.1117/12.2233913

Feinberg, L., Ziemer, J., Ansdell, M., Crooke, J., Dressing, C., Mennesson, B., O'Meara, J., Pepper, J., & Roberge, A. (2024). The Habitable Worlds Observatory engineering view: Status, plans, and opportunities. In L. E. Coyle, S. Matsuura, & M. D. Perrin (Eds.), *Space telescopes and instrumentation 2024: Optical, infrared, and millimeter wave* (Vol. 13092, p. 130921N). International Society for Optics; Photonics; SPIE. https://doi.org/10.1117/12.3018328

Stark, C. C., Roberge, A., Mandell, A., & Robinson, T. D. (2014). Maximizing the ExoEarth candidate yield from a future direct imaging mission. *The Astrophysical Journal*, *795*(2), 122. https://doi.org/10.1088/0004-637X/795/2/122

Stark, C. C., Steiger, S., Tokadjian, A., Savransky, D., Belikov, R., Chen, P., Krist, J., Macintosh, B., Morgan, R., Pueyo, L., Sirbu, D., & Stapelfeldt, K. (2025). Cross-model validation of coronagraphic exposure time calculators for the Habitable Worlds Observatory: A report from the Exoplanet Science Yield sub-Working Group. *arXiv e-Prints*, arXiv:2502.18556. https://arxiv.org/abs/2502.18556