The Journal of Open Source Software

popsynth: A generic astrophysical population synthesis framework

J. Michael Burgess[1] **and Francesca Capel**[2]

1 Max Planck Institute for Extraterrestrial Physics, Giessenbachstrasse, 85748 Garching, Germany **2** Technical University of Munich, Boltzmannstrasse 2, 85748 Garching, Germany

DOI: 10.21105/joss.03257

Software

- Review
- Repository
- Archive

Editor: Juanjo Bazán

Reviewers:

- @HeloiseS
- @warrickball

Submitted: 09 April 2021
Published: 17 July 2021

Summary

Simulating a survey of fluxes and redshifts (distances) from an astrophysical population is a routine task. `popsynth` provides a generic, object-oriented framework to produce synthetic surveys from various distributions and luminosity functions, apply selection functions to the observed variables and store them in a portable (HDF5) format. Population synthesis routines can be constructed either using classes or from a serializable YAML format allowing flexibility and portability. Users can not only sample the luminosity and distance of the populations, but they can create auxiliary distributions for parameters which can have arbitrarily complex dependencies on one another. Thus, users can simulate complex astrophysical populations which can be used to calibrate analysis frameworks or quickly test ideas.

Statement of need

`popsynth` provides a generic framework for simulating astrophysical populations with an easily extensible class inheritance scheme that allows users to adapt the code to their own needs. As understanding the rate functions of astrophysical populations (e.g., gravitational wave sources, gamma-ray bursts, active galactic nuclei) becomes an increasingly important field (Loredo & Hendry, 2019), researchers develop various ways to estimate these populations from real data. `popsynth` provides a way to calibrate these analysis techniques by producing synthetic data where the inputs are known (e.g. Mortlock et al., 2019). Moreover, selection effects are an important part of population analysis and the ability to include this property when generating a population is vital to the calibration of any survey analysis method which operates on an incomplete sample.

Similar frameworks exist for simulating data from specific catalogs such as `SkyPy` (Amara & al., 2021) and `firesong` (Tung et al., 2021), however, these have much more focused applications and do not include the ability to impose selection functions.

Procedure

Once a rate function and all associated distributions are specified in `popsynth`, a numeric integral over the rate function produces the total rate of objects in the populations. A survey is created by making a draw from a Poisson distribution with mean equal to the total rate of objects multiplied by survey duration for the number of objects in the survey. For each object, the properties such as distance and luminosity are sampled from their associated distributions. Selection functions are then applied to latent or observed variables as specified by the user.

Finally, all population objects and variables are returned in an object that can be serialized to disk for later examination. Further details on the mathematics, procedure, and details on customization can be found in the extensive documentation.

Acknowledgments

This project was inspired by conversations with Daniel J. Mortlock wherein we tried to calibrate an analysis method we will eventually get around to finishing. Inspiration also came from wanting to generalize the examples from Will Farr's lecture note (Farr, 2019). J. Michael Burgess acknowledges support from the Alexander von Humboldt Stiftung. Francesca Capel acknowledges financial support from the Excellence Cluster ORIGINS, which is funded by the Deutsche Forschungsgemeinschaft (DFG, German Research Foundation) under Germany's Excellence Strategy - EXC-2094-390783311.

References

Amara, A., & al., et. (2021). SkyPy: A package for modelling the universe. In *GitHub repository*. GitHub. https://github.com/skypyproject/skypy

Farr, W. (2019). An example of treating selection effects via summing over non-detections in stan. In *GitHub repository*. GitHub. https://github.com/farr/SelectionExample

Loredo, T. J., & Hendry, M. A. (2019). Multilevel and hierarchical Bayesian modeling of cosmic populations. *arXiv*, *astro-ph.IM*. arXiv.org

Mortlock, D. J., Feeney, S. M., Peiris, H. V., Williamson, A. R., & Nissanke, S. M. (2019). Unbiased Hubble constant estimation from binary neutron star mergers. *Physical Review D*, *100*(10), 103523. https://doi.org/10.1103/physrevd.100.103523

Tung, C. F., Glauch, T., Larson, M., Pizzuto, A., Reimann, R., & Taboada, I. (2021). FIRESONG: A python package to simulate populations of extragalactic neutrino sources. *Journal of Open Source Software*, *6*(61), 3194. https://doi.org/10.21105/joss.03194