The Journal of Open Source Software

DOI: DOIunavailable

Software

- Review
- Repository
- Archive

Editor: Pending Editor

Reviewers:

- @Pending Reviewers

Submitted: N/A
Published: N/A

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

Episimlab: A Python package for modeling epidemics

Ethan Ho[1] **and Kelly Pierce**[1]

1 Texas Advanced Computing Center (TACC) - University of Texas at Austin

Summary

Computational models play a crucial role in our scientific understanding of, and preparedness for, infectious disease epidemics. These in silico disease models simulate real-world transmission dynamics, and are thereby well-suited for such tasks as early detection of novel pandemics, improving situational awareness during periods of high prevalence, and estimating efficacy of intervention strategies. These modeling approaches have proven valuable in responding to emergent epidemics such as the H1N1 flu pandemic, the Ebola epidemic, the Zika virus pandemic, and the recent COVID-19 pandemic (ref). For example, during the early waves of the COVID-19 pandemic, compartmental disease models were instrumental for projecting case counts and hospitalizations K. A. Pierce et al. (2020). As more data on case incidence, hospitalizations, and viral genomics became available, disease modelers are able to simulate increasingly complex disease dynamics. Developing such complex compartmental models is time-consuming, however, and few implementations share a common software framework or application flow. Therefore, there is a urgent need for more robust cyberinfrastructure in the field of epidemic modeling.

`Episimlab` is a software development kit (SDK) written in Python that seeks to address this problem by providing a flexible framework for developing compartmental disease models. Models in Episimlab are collections of modular components, known as `processes`, which can be added, removed, or replaced to modify the dynamics of the simulated disease. In practice, this means that Episimlab supports development of models that:

1. Have many parameters, often with multiple, varying dimensions
2. Use any compartment structure that can be represented as a graph
3. Simulate interventions dynamically, such as administering vaccines only when case incidence exceeds a threshold.

The package is designed to be approachable; it includes several pre-packaged models that the user can run in a few lines of code. When the user chooses to add data streams or more complex transmission dynamics, they can easily do so by adding or replacing `processes` in the model. In addition, Episimlab provides a scalable and performant runtime for model execution, thanks to integration with packages in the PyData stack such as Dask (Dask Development Team, 2016), Xarray (Hoyer & Hamman, 2017), and `xarray-simlab` (Bovy et al., 2021). Finally, Episimlab provides a standard for packaging, versioning, and sharing models, using Python's built-in class attributes.

Statement of Need

`Episimlab` is a Python package that accelerates development of complex compartmental disease models. It provides sufficient boilerplate such that the user can quickly instantiate a basic compartmental model such as an SIR model (Kermack & McKendrick, 1927). Simple models such as the SIR model are not unique to `Episimlab`; they have a long history of use in epidemiology since their origin in the early 20th century Ross & Hudson (1917b). More recently, various implementations of compartmental models have been made publicly available and easily usable as open-source software packages (refs). These packages are valuable because they simplify execution of many simple compartmental models, but their usage is limited to the discrete handful of model structures that are published with the package. In addition, such projects rarely support complex models with more than 5 or 6 compartments, presumably because complex models are difficult to develop and reproduce. While inspired by these previous works, `Episimlab` aims to support development of models with arbitrary complexity. It gives the user flexibility to define key components of their compartmental model, such as the dimensionality of the Markov state matrix, the number of compartments, and custom stochastic behavior.

`Episimlab` was designed in collaboration with data scientists and epidemiologists from the UT Austin COVID-19 Modeling Consortium (ref). Specifically, prototypes of `Episimlab` were used in studies forecasting hospital burden due to the COVID-19 epidemic in Austin, Texas (refs). Although the package was originally designed with specific epidemiological use cases in mind, `Episimlab` is meant to be used by anyone developing compartmental disease models. The package is useful for students, since it provides a minimal, approachable boilerplate for developing basic models in pure Python. It introduces and reinforces best practices in object-oriented software development, such as modularity and reproducibility. For disease modeling experts, `Episimlab` provides a platform that supports a wide variety of modeling use cases. It leverages concurrency in `xarray-simlab`, dataset chunking in `Dask`, and accelerated matrix math in `xarray`, so `Episimlab` models are performant even when using large datasets.

Unused Text

Unlike past compartmental modeling software, models in `Episimlab` are modular. `Episimlab` models are essentially unique sets of discrete components. These components, formally known as `processes` in the API, can be modified, exchanged, reused, and shared between models. Commonly used `processes`, such as calculating the force of infection (FOI), ship with Episimlab, and can be modified using class inheritance in Python.

References

Bovy, B., McBain, G. D., Gailleton, B., & Lange, R. (2021). *Benbovy/xarray-simlab: 0.5.0* (Version 0.5.0) [Computer software]. Zenodo. https://doi.org/10.5281/zenodo.4469813

Dask Development Team. (2016). *Dask: Library for dynamic task scheduling*. https://dask.org

Hoyer, S., & Hamman, J. (2017). Xarray: N-D labeled arrays and datasets in Python. *Journal of Open Research Software*, *5*(1). https://doi.org/10.5334/jors.148

Kermack, W. O., & McKendrick, A. G. (1927). A contribution to the mathematical theory of epidemics. *Proceedings of the Royal Society of London. Series A, Containing Papers of a Mathematical and Physical Character*, *115*(772), 700–721. https://doi.org/10.1098/rspa.1927.0118

Pierce, K. A., Ho, E., Wang, X., Pasco, R., Du, Z., Zynda, G., Song, J., Wells, G., Fox, S. J., & Meyers, L. A. (2020). Early COVID-19 pandemic modeling: Three compartmental model case studies from texas, USA. *Computing in Science & Engineering*, *23*(1), 25–34. https://doi.org/10.1109/MCSE.2020.3037033

Pierce, K., Ho, E., Wang, X., Pasco, R., Du, Z., Fox, S., Zynda, G., Song, J., & Meyers, L. A. (2020). COVID-19 healthcare demand projections: Beaumont-port arthur MSA texas. *UT COVID-19 Model. Consortium*. https://doi.org/10.13140/RG.2.2.14066.53443

Ross, R. (1916). An application of the theory of probabilities to the study of a priori pathometry.&#x2014;part i. *Proceedings of the Royal Society of London. Series A, Containing Papers of a Mathematical and Physical Character*, *92*(638), 204–230. https://doi.org/10.1098/rspa.1916.0007

Ross, R., & Hudson, H. P. (1917a). An application of the theory of probabilities to the study of a priori pathometry.&#x2014;part II. *Proceedings of the Royal Society of London. Series A, Containing Papers of a Mathematical and Physical Character*, *93*(650), 212–225. https://doi.org/10.1098/rspa.1917.0014

Ross, R., & Hudson, H. P. (1917b). An application of the theory of probabilities to the study of a priori pathometry.&#x2014;part III. *Proceedings of the Royal Society of London. Series A, Containing Papers of a Mathematical and Physical Character*, *93*(650), 225–240. https://doi.org/10.1098/rspa.1917.0015