

# Kepler Mapper: A flexible Python implementation of the Mapper algorithm.

Hendrik Jacob van Veen<sup>1</sup>, Nathaniel Saul<sup>2</sup>, David Eargle<sup>3</sup>, and Sam W. Mangham<sup>4</sup>

<sup>1</sup> Nubank <sup>2</sup> Department of Mathematics and Statistics, Washington State University Vancouver <sup>3</sup> Leeds School of Business, University of Colorado Boulder <sup>4</sup> Department of Electronics & Computer Science, University of Southampton, Southampton, SO17 1BJ, UK

DOI: [10.21105/joss.01315](https://doi.org/10.21105/joss.01315)

## Software

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

**Submitted:** 06 March 2019

**Published:** 17 October 2019

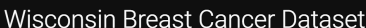
## License

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

## Summary

Topological data analysis (TDA) is a field of analysis focused on understanding the shape and structure of complex data. By computing topological descriptors of data, such as connected components, loops, and voids, we are better able to find hidden relationships among noisy and high-dimensionality data (Carlsson, 2009; Edelsbrunner & Harer, 2010). However, raw TDA metrics cannot be readily visualized. To address this gap, Singh et al. (2007) developed the Mapper algorithm to facilitate graphical exploration of topological data structures. The work of Lum et al. (2013) sparked widespread interest in the Mapper technique by demonstrating its use in multiple domains, including political science, biology, and sports analytics.

This library, Kepler Mapper, is a Python implementation of the Mapper algorithm as first described in the paper “Topological Methods for the Analysis of High Dimensional Data Sets and 3D Object Recognition” (Singh et al., 2007). Kepler Mapper presents an intuitive interface for the Mapper algorithm, and provides multiple comprehension methods for visualizing the network graph that Mapper produces. We leverage Scikit-Learn-API-compatible clustering and scaling algorithms to construct network graphs in a flexible and user-friendly way. We also provide an extensive suite of tutorials detailing the use of Kepler Mapper for simple and complex use cases.



**Figure 1:** Example Kepler Mapper graph visualization using the Wisconsin Breast Cancer Dataset (Dua & Graff, 2017)

## Library Details

Kepler Mapper provides an object-oriented API for constructing lenses and building Mapper network graphs. The module employs the [strategy pattern](#), giving users control over the clustering algorithm, covering scheme, and nerve scheme. This allows the module to be flexible for many use cases. Clustering strategies follow the Scikit-Learn clusterer interface (Pedregosa et al., 2011). We provide similar interfaces for Cover classes and Nerve classes, as well as default implementations of those classes that are most commonly found in the literature.

Visual exploration is a critical aspect of Mapper network graph analysis. For this, we provide multiple methods of visualization. For interactive visualization and exploration in the browser, Kepler Mapper can create a visual HTML interface utilizing [D3.js](#). For use with Jupyter (Kluyver et al., 2016) or other embedded purposes, we provide a visualization interface utilizing Plotly (Plotly Technologies Inc., 2015). For static visualizations, we provide an adapter so that visualization functionality from NetworkX (Hagberg, Schult, & Swart, 2008) and Matplotlib (Hunter, 2007) can be used.

## Acknowledgements

Nathaniel Saul was partially supported by NSF DBI-1661348 and by Washington NASA Space Grant Consortium, NASA Grant #NNX15AJ98H.

## References

- Carlsson, G. (2009). Topology and data. *Bulletin of the American Mathematical Society*, 46(2), 255–308. doi:[10.1090/S0273-0979-09-01249-X](https://doi.org/10.1090/S0273-0979-09-01249-X)

- Dua, D., & Graff, C. (2017). UCI machine learning repository. University of California, Irvine, School of Information; Computer Sciences. Retrieved from <http://archive.ics.uci.edu/ml>
- Edelsbrunner, H., & Harer, J. (2010). *Computational topology: An introduction*. American Mathematical Soc. doi:[10.1007/978-3-540-33259-6\\_7](https://doi.org/10.1007/978-3-540-33259-6_7)
- Hagberg, A. A., Schult, D. A., & Swart, P. J. (2008). Exploring network structure, dynamics, and function using networkx. In G. Varoquaux, T. Vaught, & J. Millman (Eds.), *Proceedings of the 7th python in science conference* (pp. 11–15). Pasadena, CA USA.
- Hunter, J. D. (2007). Matplotlib: A 2D graphics environment. *Computing in Science & Engineering*, 9(3), 90–95. doi:[10.1109/MCSE.2007.55](https://doi.org/10.1109/MCSE.2007.55)
- Kluyver, T., Ragan-Kelley, B., Pérez, F., Granger, B., Bussonnier, M., Frederic, J., Kelley, K., et al. (2016). Jupyter notebooks – a publishing format for reproducible computational workflows. (F. Loizides & B. Schmidt, Eds.). IOS Press.
- Lum, P. Y., Singh, G., Lehman, A., Ishkanov, T., Vejdemo-Johansson, M., Alagappan, M., Carlsson, J. G., et al. (2013). Extracting insights from the shape of complex data using topology. *Scientific Reports*, 3(1236). doi:[10.1038/srep01236](https://doi.org/10.1038/srep01236)
- Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., et al. (2011). Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12, 2825–2830.
- Plotly Technologies Inc. (2015). Collaborative data science. Montreal, QC: Plotly Technologies Inc. Retrieved from <https://plot.ly>
- Singh, G., Memoli, F., & Carlsson, G. (2007). Topological Methods for the Analysis of High Dimensional Data Sets and 3D Object Recognition. In M. Botsch, R. Pajarola, B. Chen, & M. Zwicker (Eds.), *Proceedings of the symposium on point based graphics* (pp. 91–100). Prague, Czech Republic: Eurographics Association. doi:[10.2312/SPBG/SPBG07/091-100](https://doi.org/10.2312/SPBG/SPBG07/091-100)