The Journal of Open Source Software

`multipers` : Multiparameter Persistence for Machine Learning

David Loiseaux[1] **and Hannah Schreiber**[1]

1 Centre Inria d'Université Côte d'Azur, France

DOI: 10.21105/joss.06773

Software

- Review
- Repository
- Archive

Editor: Rocco Meli

Reviewers:

- @yossibokorbleile
- @peekxc

Submitted: 13 May 2024
Published: 31 October 2024

Summary

`multipers` is a Python library for Topological Data Analysis, focused on **Multi**parameter **Pers**istence computation and visualizations for Machine Learning. It features several efficient computational and visualization tools, with integrated, easy to use, auto-differentiable Machine Learning pipelines, that can be seamlessly interfaced with `scikit-learn` (Pedregosa et al., 2011) and `PyTorch` (Paszke et al., 2019). This library is meant to be usable for non-experts in Topological or Geometrical Machine Learning. Performance-critical functions are implemented in `C++` or in `Cython` (Behnel et al., 2011-03/2011-04), are parallelizable with `TBB` (Robison, 2011), and have `Python` bindings and interface. It can handle a very diverse range of datasets that can be framed into a (finite) multi-filtered simplicial or cell complex, including, e.g., point clouds, graphs, time series, images, etc.

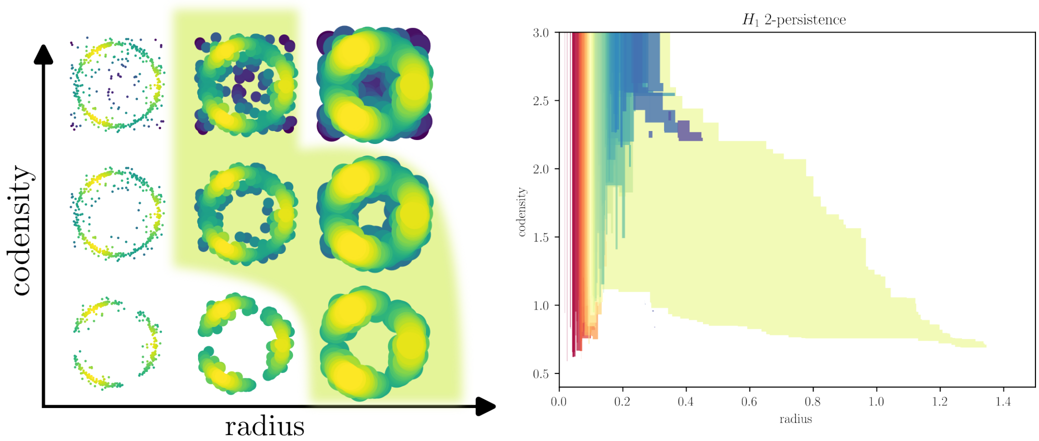

Figure 1: (Left) Topological 2-filtration grid. The color corresponds to the density estimation of the sampling measure of the point cloud. More formally, a point $x \in \mathbb{R}^2$ belongs to the grid cell with coordinates (r, d) iff $d(x, \text{point cloud}) \leq r$ and $\text{density}(x) \geq d$. The green background shape corresponds to the lifetime of the annulus in this 2-parameter grid. **(Right)** A visualization of the lifetimes of geometric structures given by `multipers`; here each colored shape corresponds to a cycle appearing in the bi-filtration on the left, and the shape represents its lifetime. The biggest green shape on the right is the same as the one on the left.

Some motivation. In the example of Figure 1, a point cloud is given from sampling a probability measure whose mass is, for the most part, located on an annulus, with some diffuse background noise. The goal here is to recover this information in a topological descriptor. For this, the point cloud can be analyzed at some geometric scale $r > 0$ and density scale d by centering balls of radius r around each point whose density is above d, and looking at the topology induced by the union of balls. However, notice that neither a fixed geometric scale nor density scale alone can retrieve (canonically) meaningful information due to the diffuse noise in the background; which is the main limitation of the prevalent approach. Nevertheless, by considering *all* possible combinations of geometric or density scales, also called a bi-filtration, it becomes straightforward with `multipers` to retrieve some of the underlying geometrical structures without relying on any arbitrary scale choice.

Furthermore, `multipers` seamlessly integrates several `Rust` and `C++` libraries such as `Gudhi` (TheGudhiProject, 2023), `filtration-domination` (Alonso et al., 2023), `mpfree` (Kerber & Rolle, 2020), and `function-delaunay` (Alonso et al., 2024), and leverages on state-of-the-art Machine Learning libraries for fast computations, such as `scikit-learn` (Pedregosa et al., 2011), `Python Optimal Transport` (Flamary et al., 2021), `PyKeops` (Charlier et al., 2021), or `PyTorch` (Paszke et al., 2019). This makes `multipers` a very efficient and fully-featured library, showcasing a wide variety of mathematically-grounded multiparameter topological invariants, including, e.g., Multiparameter Module Approximation (Loiseaux et al., 2022), Euler, Hilbert, and Rectangle Signed Barcodes (Botnan et al., 2022) (Oudot & Scoccola, 2024), Multiparameter Persistent Landscapes (Vipond, 2020); each of them computable from several multi-filtrations, e.g., Rips-Density-like filtrations, Cubical, Degree-Rips, Function-Delaunay, or any k-critical multi-filtration. These topological descriptors can then directly be used in auto-differentiable Machine Learning pipelines, using the differentiability framework developed in (Scoccola et al., 2024), through several methods, such as, e.g., Decomposable Module Representations (Loiseaux, Carrière, et al., 2023), Sliced Wasserstein Kernels or Convolutions from Signed Measures (Loiseaux, Scoccola, et al., 2023). As a result, `multipers` is capable of handling, within a single minute of computation, datasets of $\sim 50k$ points with only 5 lines of Python code. See Figures 2, 3.

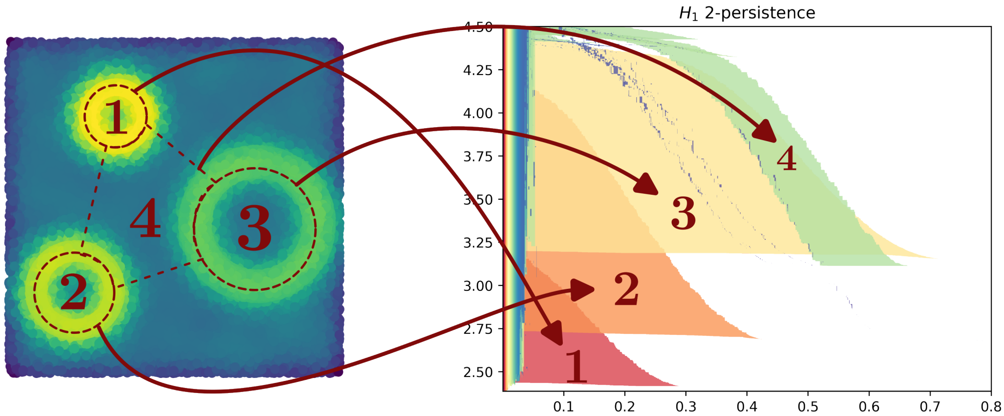

Figure 2: Typical interpretation of a "Geometric & Density" bi-filtration with `multipers`. **(Left)** Point cloud with color induced by density estimation (same as Figure 1). **(Right)** A visualization of the topological structure lifetimes computed from a Delaunay-Codensity bi-filtration; here the three cycles can be retrieved using their radii (x-axis) and their co-densities (y-axis). The first cycle is the densiest, and smallest, and thus corresponds to the one that appears in the bottom(high-density)-left(small-radius) of the bi-filtration. The second is less dense (thus above the first one) and bigger (thus more on the right). The same goes for the last one.

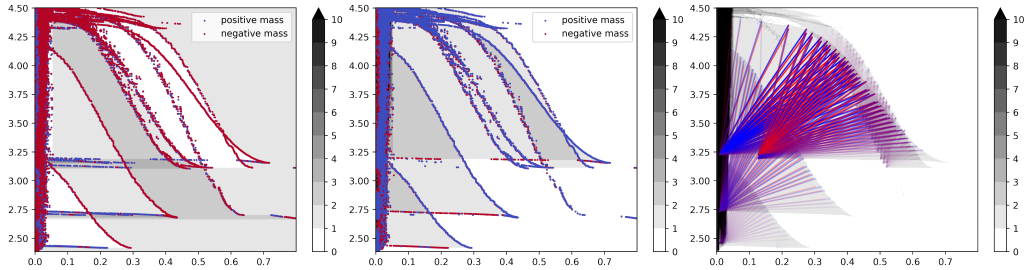

Figure 3: Different Signed Barcodes from the same dataset as Figure 1. **(Left)** Euler Decomposition Signed Barcode, and the Euler Surface in the background. **(Middle)** Hilbert Decomposition Signed Barcode, with its Hilbert Function surface. **(Right)** Rank invariant Signed Barcode, with the Hilbert Function as a background.

The core functions of the Python library are automatically tested on Linux and macOS, using `pytest` (Krekel et al., 2004) alongside GitHub Actions.

Related work and statement of need

There exists several libraries for computation or pre-processing of very specific tasks related to multiparameter persistence. However, to the best of our knowledge, none of them are able to tackle the challenges that `multipers` is dealing with, i.e., **(1)** computing and unifying the computations of multiparameter persistent structures, in a non-expert friendly approach, and **(2)** provide ready-to-use general tools to use these descriptors for Machine Learning pipelines and projects.

Eulearning. This library features different approaches for computing and using the Euler Characteristic of a multiparameter filtration (Hacquard & Lebovici, 2023). Although relying on distinct methods, `multipers` can also be used to compute Machine Learning descriptors from the Euler Characteristic, i.e., the Euler Decomposition Signed Barcode, or Euler Surfaces. Moreover, `multipers` computations are faster (especially on point cloud datasets), easier to use, and available on a wider range of multi-filtrations.

Multiparameter Persistent Landscape. Implemented on top of Rivet (Lesnick & Wright, 2015), this library computes a multiparameter persistent descriptor by computing 1-parameter persistence landscape vectorizations of slices in multi-filtrations (Vipond, 2020), called Multiparameter Persistent Landscape (MPL). This library also features some multiparameter persistence visualizations. However, it is limited to `Rivet` capabilities and landscapes computations, which on one hand does not leverage on recently developed optimizations, e.g., (Alonso et al., 2023), or (Kerber & Rolle, 2020), and on the other hand can only work with very specific text file inputs.

GRIL. This library provides code to compute a specific, generalized version of the Multiparameter Persistent Landscapes (Xin et al., 2023), relying on 1-paramter persistence zigzag computations. This library however is limited to this invariant, can only deal with 2-parameter persistence, and is not as much integrated as `multipers` with other multiparameter persistence and Machine Learning libraries.

Elder Rule Staircode. This library features a descriptor for 2-parameter, degree-0 homology, rips-densitity-like filtrations (Cai et al., 2021). Once again, this library is very specific and not linked with other libraries.

Persistable. is a GUI interactive library for clustering, using degree-0 multiparameter persistence (Scoccola & Rolle, 2023) (Rolle & Scoccola, 2020). Although aiming at distinct goals and using very different approaches, `multipers` can also be used for clustering, by computing (differentiable) descriptors that can be used afterward with standard clustering methods, e.g., K-means.

We contribute to this variety of task-specific libraries by providing a **general purpose** library, `multipers`, with novel and efficient topological invariant computations, integrated state-of-the art Machine Learning topological pipelines, and interfaces to standard Machine Learning and Deep Learning libraries.

Acknowledgements

David Loiseaux was supported by ANR grant 3IA Côte d'Azur (ANR-19-P3IA-0002). The authors would like to thank Mathieu Carrière, and Luis Scoccola for their help on Sliced Wasserstein, and Möbius inversion code.

References

Alonso, Á. J., Kerber, M., Lam, T., & Lesnick, M. (2024). Delaunay Bifiltrations of Functions on Point Clouds. In *Proceedings of the 2024 Annual ACM-SIAM Symposium on Discrete Algorithms (SODA)* (pp. 4872–4891). Society for Industrial and Applied Mathematics. https://doi.org/10.1137/1.9781611977912.173

Alonso, Á. J., Kerber, M., & Pritam, S. (2023). Filtration-domination in bifiltered graphs. In *2023 proceedings of the symposium on algorithm engineering and experiments (ALENEX)* (pp. 27–38). https://doi.org/10.1137/1.9781611977561.ch3

Behnel, S., Bradshaw, R., Citro, C., Dalcin, L., Seljebotn, D. S., & Smith, K. (2011-03/2011-04). Cython: The best of both worlds. *Computing in Science Engineering*, *13*(2), 31–39. https://doi.org/10.1109/MCSE.2010.118

Botnan, M. B., Oppermann, S., & Oudot, S. (2022). Signed Barcodes for Multi-Parameter Persistence via Rank Decompositions. In X. Goaoc & M. Kerber (Eds.), *38th International Symposium on Computational Geometry (SoCG 2022)* (Vol. 224, pp. 19:1–19:18). Schloss Dagstuhl – Leibniz-Zentrum für Informatik. https://doi.org/10.4230/LIPIcs.SoCG.2022.19

Cai, C., Kim, W., Memoli, F., & Wang, Y. (2021). Elder-Rule-Staircodes for Augmented Metric Spaces. *SIAM Journal on Applied Algebra and Geometry*, *5*(3), 417–454. https://doi.org/10.1137/20M1353605

Charlier, B., Feydy, J., Glaunès, J. A., Collin, F. ois-D., & Durif, G. (2021). Kernel Operations on the GPU, with Autodiff, without Memory Overflows. *Journal of Machine Learning Research*, *22*(74), 1–6.

Flamary, R., Courty, N., Gramfort, A., Alaya, M. Z., Boisbunon, A., Chambon, S., Chapel, L., Corenflos, A., Fatras, K., Fournier, N., Gautheron, L., Gayraud, N. T. H., Janati, H., Rakotomamonjy, A., Redko, I., Rolet, A., Schutz, A., Seguy, V., Sutherland, D. J., … Vayer, T. (2021). POT: Python optimal transport. *The Journal of Machine Learning Research*, *22*(1), 78:3571–78:3578.

Hacquard, O., & Lebovici, V. (2023). Euler Characteristic Tools For Topological Data Analysis. *arXiv.org*. https://doi.org/10.48550/arxiv.2303.14040

Kerber, M., & Rolle, A. (2020). Fast Minimal Presentations of Bi-graded Persistence Modules. *arXiv:2010.15623 [Cs, Math]*. https://doi.org/10.1137/1.9781611976472.16

Krekel, H., Oliveira, B., Pfannschmidt, R., Bruynooghe, F., Laugher, B., & Bruhin, F. (2004). *Pytest 8.3*.

Lesnick, M., & Wright, M. (2015). Interactive visualization of 2-D persistence modules. *arXiv:1512.00180 [Cs, Math]*. https://doi.org/10.48550/arXiv.1512.00180

Loiseaux, D., Carrière, M., & Blumberg, A. (2023). A Framework for Fast and Stable Representations of Multiparameter Persistent Homology Decompositions. *Advances in Neural Information Processing Systems*, *36*, 35774–35798.

Loiseaux, D., Carrière, M., & Blumberg, A. J. (2022). *Fast, Stable and Efficient Approximation of Multi-parameter Persistence Modules with MMA*. https://doi.org/10.48550/arXiv.2206.02026

Loiseaux, D., Scoccola, L., Carrière, M., Botnan, M. B., & Oudot, S. (2023). Stable Vectorization of Multiparameter Persistent Homology using Signed Barcodes as Measures. *Advances in Neural Information Processing Systems*, *36*, 68316–68342.

Oudot, S., & Scoccola, L. (2024). On the Stability of Multigraded Betti Numbers and Hilbert Functions. *SIAM Journal on Applied Algebra and Geometry*, *8*(1), 54–88. https://doi.org/10.1137/22M1489150

Paszke, A., Gross, S., Massa, F., Lerer, A., Bradbury, J., Chanan, G., Killeen, T., Lin, Z., Gimelshein, N., Antiga, L., Desmaison, A., Köpf, A., Yang, E., DeVito, Z., Raison, M., Tejani, A., Chilamkurthy, S., Steiner, B., Fang, L., … Chintala, S. (2019). PyTorch: An imperative style, high-performance deep learning library. In *Proceedings of the 33rd International Conference on Neural Information Processing Systems* (pp. 8026–8037). Curran Associates Inc.

Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., Vanderplas, J., Passos, A., Cournapeau, D., Brucher, M., Perrot, M., & Duchesnay, É. (2011). Scikit-learn: Machine Learning in Python. *Journal of Machine Learning Research*, *12*(85), 2825–2830.

Robison, A. D. (2011). Intel® Threading Building Blocks (TBB). In D. Padua (Ed.), *Encyclopedia of Parallel Computing* (pp. 955–964). Springer US. https://doi.org/10.1007/978-0-387-09766-4_51

Rolle, A., & Scoccola, L. (2020). Stable and consistent density-based clustering via multiparameter persistence. *arXiv.org*. https://doi.org/10.48550/arXiv.2005.09048

Scoccola, L., & Rolle, A. (2023). Persistable: Persistent and stable clustering. *Journal of Open Source Software*, *8*(83), 5022. https://doi.org/10.21105/joss.05022

Scoccola, L., Setlur, S., Loiseaux, D., Carrière, M., & Oudot, S. (2024). Differentiability and Optimization of Multiparameter Persistent Homology. *Proceedings of the 41st International Conference on Machine Learning*, *235*, 43986–44011.

TheGudhiProject. (2023). *GUDHI*. GUDHI Editorial Board.

Vipond, O. (2020). Multiparameter persistence landscapes. *Journal of Machine Learning Research*, *21*, 61:1–61:38.

Xin, C., Mukherjee, S., Samaga, S. N., & Dey, T. K. (2023). GRIL: A 2-parameter Persistence Based Vectorization for Machine Learning. *Proceedings of 2nd Annual Workshop on Topology, Algebra, and Geometry in Machine Learning (TAG-ML)*, 313–333.