Data Reduction and Feature Isolation for Computing Persistent Homology on High Dimensional Data

Rishi R. Verma, Nicholas O. Malott, and Philip A. Wilsey
Dept. of EECS, University of Cincinnati, Cincinnati, OH 45221, USA
Email: verma.rishiraj@gmail.com, malottno@mail.uc.edu, philip.wilsey@uc.edu

***Abstract*—Persistent Homology (PH) is computationally expensive and is thus generally employed with strict limits on the (i) maximum connectivity distance and (ii) dimensions of homology groups to compute (unless working with trivially small data sets). As a result, most studies with PH only work with H_0 and H_1 homology groups. This paper examines the identification and isolation of regions of data sets where high dimensional topological features are suspected to be located. These regions are analyzed with PH to characterize the high dimensional homology groups contained in that region. Since only the region around a suspected topological feature is analyzed, it is possible to identify high dimension homologies piecewise and then assemble the results into a scalable characterization of the original data set.**

***Index Terms*—High Dimensional Data; Data Reduction; Persistent Homology; Involuted Homology; Data Mining**

I. Introduction

Persistent Homology (PH) is one of the principal components of *Topological Data Analysis (TDA)* [1]. PH provides a scalar representation of topological structures encoded in a point cloud — connected components, loops, voids, and so on — as they persist over multiple spatial resolutions [2]–[6]. The output of PH is captured through *persistence intervals* that characterize the persistence, or spatial lifetime, of these topological structures. The resulting persistence intervals can be used for various machine learning applications [7]–[18].

The computation of PH suffers from exponential memory and time complexities that prevent its general application to high dimensional data. For example, computing the H_2 homology groups of a point cloud in $\mathbb{R}^3$ is only feasible with a few thousand points. Consequently, work with PH is often limited to the H_0 and H_1 homology groups (although in some cases H_2 homology groups are studied). However, topological features in dimensions above H_2 are generally omitted due to the associated time and space complexity. This work explores techniques to enable the use of computational PH to characterize higher dimensional homology groups.

The method presented in this paper computes PH on dimensional reductions of the data to locate candidate regions where high dimensional homologies may be located. The candidate regions of the original space are then isolated for high dimension PH analysis. The technique is motivated by previous studies of dimensionality reduction on PH [19]–[22]. The principal idea of this paper is that, under projection, higher dimensional topological features will be present in the lower dimensional data. Thus, computing PH on the dimensionality reduced data will help identify candidate regions of the original data to examine for high dimensional homologies. In this work, dimensionality reduction and PH are used to locate the vertices of representative cycles of topological features in the reduced dimension. The dimensionality reduced vertices are then re-associated to the corresponding vertices in the original data set. The vertex set is then used to extract a region of the original data around the cycle's center for further PH analysis in the high dimensional space.

The remainder of this paper is organized as follows. Section II presents some background and related work. Section III reviews the technical approach used to locate and analyze topological features in high dimensional data. Section IV presents some early experimental results. Finally, Section V contains some concluding remarks.

II. Background and Related Work

This section presents background information and studies related to this work. Additional background information on PH can be found in [2], [4], [23], [24].

Techniques to combat the exponential memory and time complexity of PH have led to advancements in complex construction [25]–[27], complex representation [28]–[30], boundary matrix reduction [30]–[33], and approximate algorithms [20], [21], [34]–[36]. A majority of these are evaluated with respect to low dimensional topological feature identification. Few studies are directly aimed at analyzing high dimensional homology groups; both the application and understanding of higher dimensional homology groups are limited.

Several studies have examined the application of dimensional reduction techniques to the computation of PH [19]–[21]. The studies by [19], [21] have largely focused on the theoretical issues and limits of dimensionality reduction for the computation of PH. Ramamurthy *et al* [20] provide the first empirical study of random projection on persistence intervals and Betti numbers. Their experiments consider high dimensional source data in 3 studies, namely: $10,000$ points in 50-dimensions (synthetic), $15,000$ points in 25-dimensions (image patches), and $4,000$ points in 100-dimensions (audio clips of wheezing patients). The dimensional reduction step

Support for this work was provided in part by the National Science Foundation under grant IIS–1909096.

978-1-6654-3902-2/21/$31.00 ©2021 IEEE

H_d	H_2	H_3	H_4	H_5	H_6	H_7	H_8
n_{max}	5164	1125	372	170	102	77	64
H_d	H_9	H_{10}	H_{11}	H_{12}	H_{13}	H_{14}	H_{15}
n_{max}	61	57	50	49	53	53	32

TABLE I: The experimentally-determined maximum number of vertices of a d-Sphere that an AMD Ryzen 7 with 128GB RAM can analyze with the `ripserer` package when PH was computed up to H_d with no distance limits.

is performed using random projection and the data is reduced to dimension 30. They examined the impact of dimensional reduction on the computed *Betti numbers* (the number of k-dimensional features, using PH). Unfortunately, the data in these studies contained Betti numbers only at β_0, β_1, and β_2 (β_2 results were reported only from the synthetic data). This study provides evidence that homology groups H_0, H_1, and H_2 are preserved through random projection; however the preservation of homology groups above H_2 was not explored.

Cycles and Co-cycles: In this paper, the identification of high dimensional homology groups is achieved by extracting *representative cycles* from a PH computation. For each topological feature, the PH computation can track the boundary, or the vertices, that circumscribe that feature [37]. There may exist multiple possible cycles that can generate the same feature, so a single *representative cycle* is chosen. The representative cycles provide candidate regions for analysis in the original space. Features identified in the projected space are utilized to gather their corresponding high dimensional neighborhood (region) that is then used to compute high dimensional homology groups (generally from a memory-bound approximated sampling of the region).

Unfortunately, most existing tools to compute PH utilize cohomology instead of homology [38]. Cohomology reports representative co-cycles instead of cycles. A new technique called *Involuted Homology* [39] can reconstruct the homology cycles after computing the cohomology. While not widely used, involuted homology is available in a tool called `ripserer` [40]. This tool is used in the studies of this paper.

Data Size Limitations when Computing PH: The approximation of high dimensional homology groups developed in this study is motivated primarily by the limits of memory. In general, computing the H_2 homology groups of data with PH is constrained to several thousand points on modern computing hardware. Computation of higher dimensional homology groups is even more restricted. Table I presents the limitations of the computation of H_d homology groups using synthetic d-Spheres in $\mathbb{R}^{d+1}$ with `ripserer` on an AMD Ryzen 7 with 128GB of RAM. These limits are similar for other tools (*e.g.,* `ripser` [30]). While the exact limits depend on each specific data set, these are sufficient for the early assessments computed for this paper.

III. Technical Approach

This section describes the approach to characterizing high dimensional homology groups of an input point cloud. The approach utilizes the output of PH on a dimensionally-reduced approximation to identify candidate regions of the space where high dimensional features may lie. These areas are then re-associated to their points in the original point cloud to form a region that is evaluated for higher dimensional homologies. Algorithm 1 contains the pseudo-code for the approach. The algorithm can be summarized in 3 main processing steps.

Algorithm 1 Locating High Dimensional Features

Input origData ▷ Input point cloud
Input highDim ▷ Maximum homology dimension
Input ripsLim ▷ Max points computable in $H_{highDim}$
Input numTrials ▷ The number of trials to perform
Output outPIs ▷ PIs from each region

```
1:  regionCenters ← ∅
2:  for i ← 1 to numTrials do
3:      reducedData ← projection(origData, dim= 3)
4:      perIntervals ← ripserer(reducedData)
5:      for all pi_i ∈ perIntervals do
6:          if pi_i.length > cutoff then
7:              hdVert ← { origData[pi_i.vertexIndices] }
8:              regionCenters.append(geometric_center(hdVert))
9:          end if
10:     end for
11: end for
12: centroids ← cluster(regionCenters).centroids
13: for all center_i ∈ centroids do
14:     radius ← μ + σ of the distance between all hdVert of the
                 cluster and the center_i
15:     region ← {pt ∈ origData | dist(pt, center_i) ≤ radius}
16:     reducedRegion ← k-means++(region, k = ripsLim)
17:     outPIs ← ripserer(reducedRegion)
18: end for
```

Step 1: Project Data to $\mathbb{R}^3$ and Compute PH

This step locates the geometric center of regions in the data where topological features are suspected to exist (Lines 2–11). Since the projection methods being used in this study are stochastic, the algorithm takes multiple trials to collect the persistence intervals in the low dimension space (this study used Gaussian projection to a fixed dimension of 3 with 8 trials). The algorithm only collects significant H_1 and H_2 persistence intervals from the projected data (Lines 5–10). In this study, the *kneed* algorithm [41] with the default sensitivity of 1.0 is used on the length ($\epsilon_{death} - \epsilon_{birth}$) of the persistence intervals to determine significance. Furthermore, the kneed algorithm and the definition of significance was evaluated separately on the sets of H_1 and H_2 persistence intervals.

For each significant persistent interval found, the indices of the representative cycle vertices are used to collect the corresponding vertices in the high dimensional data (Line 7). The index is sufficient to re-associate vertices back to the high dimensional space because Gaussian projection performs a 1-to-1 mapping of high to low vectors. The final component of this step is to compute the geometric center of the high dimension vertices associated to the persistence interval (Line 8). Note that all geometric centers are collected regardless of the dimension of the persistence interval that generated them.

A quick evaluation to analyze each dimension separately was performed early in this study, but no significant advantage or discriminating conclusion could be drawn from their separation.

Step 2: Cluster the Centers and Extract Centroids

In this step, the geometric centers, gathered from the representative cycles in Step 1, are now examined and clustered to *isolate the key regions* where topological features are suspected to exist. In particular, the geometric centers are clustered and the centroids of the clusters captured (Line: 12). This approach requires a clustering algorithm that operates without expecting a known target number of clusters. In this study, the MeanShift algorithm was used. While not scalable, the number of representative cycles to cluster is relatively small and the MeanShift algorithm performs extremely well experimentally at locating the correct number of centers for the test data examined in Section IV. The clustering step removes redundant centers collected from the persistence intervals of the multiple trials.

Step 3: Define High Dimension Region & Compute PH

Each centroid from Step 2 is used to define the center of a candidate region for more detailed analysis in the higher dimensional space. For each centroid, points from the original point cloud that are within a specified distance of the centroid are collected into a region of points of interest (Line: 15). In this study, this distance is defined independently for each cluster. In particular, the mean (μ) plus standard deviation (σ) of the distances from the vertices of the representative cycles in the cluster to the cluster centroid is used (Line: 14). This region of points is a subset of the original point cloud that can be studied with PH. However, the size of this region can easily exceed the capacity of existing PH tools, so prior to analysis, the region is sampled (Line: 16). The k-means++ algorithm is used to sample the data as it has been shown to provide an accurate approximation for computing PH for large data sets [35], [42]. The sampled region is then analyzed using standard PH tools (Line: 17).

IV. Preliminary Results

Synthetic data sets are used to evaluate the technique described in this paper in order to generate known high dimensional features. The synthetic data is composed of multiple unit d-Spheres of different dimensions embedded in $\mathbb{R}^5$ through $\mathbb{R}^7$. Each unit d-Sphere is composed of $5,000$ points and was generated using Muller's method [43] to produce a uniform random set of points on the surface of the sphere. These spheres were all constructed about the origin of the coordinate axis in $\mathbb{R}^{d+1}$. One test was performed on data embedded in $\mathbb{R}^8$. However the computation of PH was unable to clearly distinguish features in this dimension (with this algorithm or using a standard PH computation on one unit 8-Sphere). As a result, testing was halted at $\mathbb{R}^7$.

Multiple d-Spheres were embedded in a common space; when the space was of a dimension higher than the original sphere, the additional coordinates were set to 0.0. The d-Spheres were placed adjacent to each other on the first coordinate axis at multiples of 2 so that they were pairwise non-intersecting. Thus, the t spheres were positioned so that their respective centers were at $\langle 2i, 0, \cdots, 0\rangle$ for $0 \leq i < t$. In this test suite, there were 9 sets of synthetic test data, three with 2 d-Spheres, two with 3 d-Spheres and four with 4 d-Spheres. Due to space considerations, not all results are shown; however, the results are consistent across all tests.

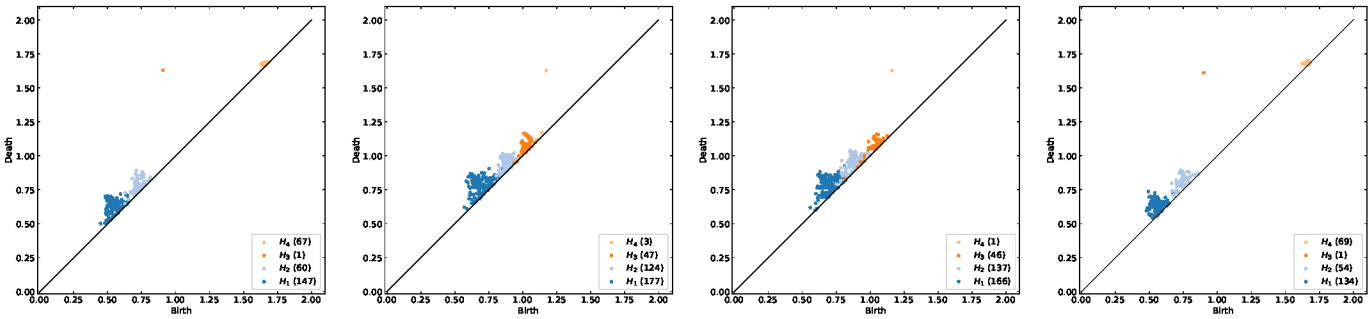
	x_0	x_1	x_2	x_3	x_4	x_5
4d5d5d4din5d: two 4-Spheres and two 5-Spheres in $\mathbb{R}^5$						
c_0 :	0.0806	0.0038	0.0295	-0.0105	0.0078	
c_1 :	2.0050	0.0142	-0.0062	0.0106	0.0008	
c_2 :	3.9756	0.0110	-0.0366	-0.0225	0.0052	
c_3 :	5.9567	0.0083	-0.0535	0.0205	0.0006	
4d5din6d: one 4-Sphere and one 5-Sphere in $\mathbb{R}^6$						
c_0 :	0.1329	-0.0253	-0.0758	0.0746	-0.0162	0.0000
c_1 :	1.8719	0.0076	-0.0051	-0.0123	0.0013	0.0000
4d6din6d: one 4-Sphere and one 6-Sphere in $\mathbb{R}^6$						
c_0 :	0.2056	0.0131	-0.0042	0.0226	0.0123	-0.0064
c_1 :	1.9366	-0.0440	-0.0339	-0.0223	-0.0139	0.0037
4d5d5d6din6d: one 4-Sphere, two 5-Spheres, and one 6-Sphere in $\mathbb{R}^6$						
c_0 :	0.0183	0.0013	0.0039	0.0124	-0.0073	0.0000
c_1 :	2.0049	0.0037	0.0189	0.0007	-0.0162	0.0000
c_2 :	4.0000	0.0031	0.0334	0.0173	0.0033	-0.0025
c_3 :	5.8807	-0.0123	0.0097	-0.0250	0.0160	0.0412

TABLE II: Centroids computed by Algorithm 1 at Line: 12.

A. Identifying Centers of Candidate regions

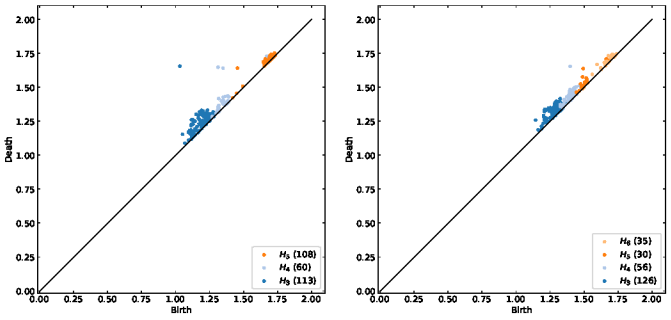
The first step of the proposed approach (Algorithm 1, Lines: 2–11) analyzes the projected data to locate centers of regions of the data that the approach determines are potential candidates where a significant topological feature may exist. This step operates without any knowledge of the number or dimension of any homological features present in the data. Given the synthetic test data described above, it is known (to us, but not to the algorithm) that there are significant topological features along the first coordinate axis at $\langle 2i, 0, \cdots, 0\rangle$ for $0 \leq i < t$. Thus, the ability of the algorithm to correctly identify these feature centers provides a preliminary indicator of the suitability of the approach.

Table II shows the computed region centers for 4 of the test data sets. The bold strings are the names of each data set, which describes the dimensions of the d-Spheres and their embeddings. For example, the first row of Table II examines **4d5d5d4din5d**. This name indicates an embedding in $\mathbb{R}^5$ of: (i) a unit 4-Sphere at $\langle 0, 0, 0, 0, 0\rangle$, (ii) a unit 5-Sphere at $\langle 2, 0, 0, 0, 0\rangle$, (iii) a unit 5-Sphere at $\langle 4, 0, 0, 0, 0\rangle$, and (iv) a unit 4-Sphere at $\langle 6, 0, 0, 0, 0\rangle$. Each row following the test data set name provides the Cartesian coordinates of the geometric centroid of all cluster centroids reported by Algorithm 1 (Line: 12). For each d-Sphere, the method reports the correct number of clusters and a good approximation of their geometric center. While not the exact centers of the test data, the computed centroids are extremely close to the correct values.



(a) 4-Sphere at $\langle 0, 0, 0, 0, 0 \rangle$ (b) 5-Sphere at $\langle 2, 0, 0, 0, 0 \rangle$ (c) 5-Sphere at $\langle 4, 0, 0, 0, 0 \rangle$ (d) 4-Sphere at $\langle 6, 0, 0, 0, 0 \rangle$

Fig. 1: Persistence Diagrams for Data Set: 4d5d5d4din5d



(a) 4d5d5d6din6d (b) 5d7d6din7d

Fig. 2: Persistence Diagrams showing homologies $> H_2$ for two test data sets.

B. Computing PH on the Candidate Regions

Step 3 of Algorithm 1 (Lines: 13–18) assembles a set of points (into a region) around the cluster centroids to analyze with PH. The approach reduces this region with KMeans++ so that it can be analyzed in the high dimensional space, then computes the PH of the sampled data. The first step determines which points from the original data set to include in the region by selecting all that are within a certain radius of the center. This radius is dynamically computed as a function of the mean radius plus standard deviation of the vertices of the representative cycles that contributed to the identification of that cluster (Line: 14). The radius used here is only one possibility and as test data with more irregular data is considered, it might be useful to use other bounding limits.

The set of persistence diagrams from test data **4d5d5d4din5d** is shown in Figure 1 (the number in parentheses in each legend entry is the number of persistence intervals at that dimension). While these results all show multiple topological features at H_3 and H_4, most of them are near the $45°$ line, which indicates noise or insignificant features. However, in each case there is one significant feature present (away from the $45°$ line). In each persistence diagram there should be one significant feature corresponding to the d-Sphere centered in that region. The results show one significant feature in the correct homology group for each d-Sphere present in the original data set.

Figure 2 contains the results from two other test data sets, namely: **4d5d5d6din6d** and **5d7d6din7d**. These persistence diagrams contain data for all of the persistence intervals found in dimensions > 2. As can be seen in the data, the left graph (a) correctly shows one significant feature at H_3 and H_5 and two significant features at H_4. The second graph (b) in Figure 2 is from an embedding in $\mathbb{R}^7$. While the significant features in H_4 and H_5 are visible, the feature at H_6 is beginning to be blurred into the noise portion of the data (closer to the $45°$ line). Part of this may be caused by the delay of the feature birth (ϵ_{birth}) caused by sampling or the high dimension of the features themselves (*cf* the shifted ϵ_{birth} for the higher dimensional features in both graphs of Figure 2).

V. Conclusions

The computation of Persistent Homology (PH) for homology groups $> H_2$ is seriously compromised due to the computational complexities of the PH algorithms. This work proposes to isolate regions in a point cloud where suspected homologies may be present and works to assemble a subset of points from the original data to form a localized region for analysis. Using this subset, the PH computation can be performed to search for higher dimension homologies. This work shows that, for a few test cases, the higher dimensional homologies can be correctly located, isolated, and approximately characterized. However, it is important to recognize that the test data (d-Spheres) is a near ideal test case for the proposed algorithm (especially w.r.t. the selection of points to form each region (Line: 14 of Algorithm 1). That said, this is still very early work and there are a number of parameters and possible modifications for Algorithm 1 that can improve and expand its performance for other topological shapes. In addition to additional testing with a more diverse set of data, other projection and clustering algorithms might prove informative.

While this result indicates an ability to approximately characterize homologies in H_3 through H_5, and, to some extent, H_6, the an important question remains: Is the ability to characterize these homology groups of any use in machine learning? On that we do not yet have any evidence.

References

[1] G. Carlsson, "Topology and data," *Bulletin of the American Mathematical Society*, vol. 46, no. 3, pp. 255–308, Apr. 2009.

[2] F. Chazal and B. Michel, "An introduction to topological data analysis: Fundamental and practical aspects for data scientists," *ArXiv e-prints*, Oct. 2017.

[3] U. Fugacci, S. Scaramuccia, F. Iuricich, and L. D. Floriani, "Persistent homology: a step-by-step introduction for newcomers." in *Smart Tools and Apps for Graphics – Eurographics Italian Chapter Conference*, G. Pintore and F. Stanco, Eds. The Eurographics Association, 2016, pp. 1–10.

[4] N. Otter, M. A. Porter, U. Tillmann, P. Grindrod, and H. A. Harrington, "A roadmap for the computation of persistent homology," *EPJ Data Science*, vol. 6, no. 1, Aug. 2017.

[5] C. S. Pun, K. Xia, and S. X. Lee, "Persistent-homology-based machine learning and its applications – a survey," Nov. 2018.

[6] X. Zhu, "Persistent homology: An introduction and a new text representation for natural language processing," in *IJCAI*, 2013, pp. 1953–1959.

[7] P. Bendich, J. S. Marron, E. Miller, A. Pieloch, and S. Skwerer, "Persistent homology analysis of brain artery trees," *The Annals of Applied Statistics*, vol. 10, no. 1, pp. 198–218, Mar. 2016.

[8] Z. Cang, L. Mu, K. Wu, K. Opron, K. Xia, and G.-W. Wei, "A topological approach for protein classification," *Molecular Based Mathematical Biology*, vol. 3, no. 1, Nov. 2015.

[9] P. G. Camara, D. I. S. Rosenbloom, K. J. Emmett, A. J. Levine, and R. Rabadan, "Topological data analysis generates high-resolution, genome-wide maps of human recombination," *Cell systems*, vol. 3, no. 1, pp. 83–94, 2016.

[10] G. Carlsson, T. Ishkhanov, V. de Silva, and A. Zomorodian, "On the local behavior of spaces of natural images," *International Journal of Computer Vision*, vol. 76, no. 1, pp. 1–12, Jan. 2008.

[11] G. Carlsson, "Topological pattern recognition for point cloud data," *Acta Numerica*, vol. 23, pp. 289–368, 2014.

[12] J. M. Chan, G. Carlsson, and R. Rabadan, "Topology of viral evolution," *Proceedings of the National Academy of Sciences*, vol. 110, no. 46, pp. 18 566–18 571, 2013.

[13] T. K. Dey and S. Mandal, "Protein classification with improved topological data analysis," in *18th International Workshop on Algorithms in Bioinformatics*, ser. WABI 2018, Aug. 2019, pp. 6:1–6:13.

[14] R. Ghrist and A. Muhammad, "Coverage and hole-detection in sensor networks via homology," in *Fourth International Symposium on Information Processing in Sensor Networks*, ser. IPSN 2005. IEEE, Apr. 2005, pp. 254–260.

[15] D. Horak, S. Maletić, and M. Rajković, "Persistent homology of complex networks," *Journal of Statistical Mechanics: Theory and Experiment*, Mar. 2009.

[16] L. Li, W.-Y. Cheng, B. S. Glicksberg, O. Gottesman, R. Tamler, R. Chen, E. P. Bottinger, and J. T. Dudley, "Identification of type 2 diabetes subgroups through topological analysis of patient similarity," *Science translational medicine*, vol. 7, no. 311, Oct. 2015.

[17] A. Moitra, N. O. Malott, and P. A. Wilsey, "Persistent homology on streaming data," in *2020 International Conference on Data Mining Workshops (ICDMW)*, ser. ICDMW '20, Nov. 2020, pp. 636–643.

[18] M. Nicolau, A. J. Levine, and G. Carlsson, "Topology based data analysis identifies a subgroup of breast cancers with a unique mutational profile and excellent survival," *Proceedings of the National Academy of Sciences*, vol. 08, no. 17, pp. 7265–7270, 2011.

[19] S. Arya, J.-D. Boissonnat, K. Dutta, and M. Lotz, "Dimensionality reduction for k-distance applied to persistent homology," in *36th International Symposium on Computational Geometry*, ser. SoCG 2020, Jun. 2020, pp. 10:1–10:15.

[20] K. N. Ramamurthy, K. R. Varshney, and J. J. Thiagarajan, "Computing persistent homology under random projection," in *IEEE Workshop on Statistical Signal Processing*, Jun. 2014, pp. 105–108.

[21] D. R. Sheehy, "The persistent homology of distance functions under random projection," in *Proceedings of the Thirtieth Annual Symposium on Computational Geometry*, ser. SOCG'14. New York, NY, USA: ACM, 2014, pp. 328–334.

[22] B. Rieck and H. Leitte, "Persistent homology for the evaluation of dimensionality reduction schemes," *Computer Graphics Forum*, 2015.

[23] H. Edelsbrunner and J. Harer, "Persistent homology — a survey," *Surveys on Discrete and Computational Geometry*, vol. 453, pp. 257–282, 2008.

[24] ——, *Computational Topology, An Introduction*. American Mathematical Society, 2010.

[25] A. Zomorodian, "Fast construction of the vietoris–rips complex," *Computer and Graphics*, pp. 263–271, 2010.

[26] J. A. Barmak and E. G. Minian, "Strong homotopy types, nerves and collapses," *Discrete & Computational Geometry*, vol. 47, no. 2, pp. 301–328, Mar. 2012.

[27] T. K. Dey, D. Shi, and Y. Wang, "Simba: An efficient tool for approximating rips-filtration persistence via simplicial batch-collapse," *24th Annual European Symposium on Algorithms (ESA 2016)*, 2016.

[28] J.-D. Boissonnat and C. Maria, "The simplex tree: An efficient data structure for general simplicial complexes," *Algorithmica*, vol. 70, no. 3, pp. 406–427, Nov. 2014.

[29] J.-D. Boissonnat, T. K. Dey, and C. Maria, "The compressed annotation matrix: an efficient data structure for computing persistent cohomology," *CoRR*, vol. abs/1304.6813, 2013. [Online]. Available: http://arxiv.org/abs/1304.6813

[30] U. Bauer, "Ripser: efficient computation of vietoris-rips persistence barcodes," 2019.

[31] M. Mrozek and B. Batko, "Coreduction homology algorithm," *Discrete & Computational Geometry*, vol. 41, no. 1, pp. 96–118, Jan. 2009.

[32] C. Chen and M. Kerber, "Persistent homology computation with a twist," in *Proceedings 27th European Workshop on Computational Geometry (EuroCG'11)*, 2011, pp. 197–200.

[33] U. Bauer, M. Kerber, and J. Reininghaus, "Clear and compress: Computing persistent homology in chunks," in *Topological Methods in Data Analysis and Visualization III*, P. T. Bremer, I. Hotz, V. Pascucci, and R. Peikert, Eds. Springer International Publishing, Mar. 2014, pp. 103–117.

[34] F. Chazal, B. T. Fasy, F. Lecci, B. Michel, A. Rinaldo, and L. Wasserman, "Subsampling methods for persistent homology," in *International Conference on Machine Learning*, ser. ICML 2015, Lille, France, Jul. 2015.

[35] A. Moitra, N. Malott, and P. A. Wilsey, "Cluster-based data reduction for persistent homology," in *2018 IEEE International Conference on Big Data*, ser. Big Data 2018, Dec. 2018, pp. 327–334.

[36] V. de Silva and G. Carlsson, "Topological estimation using witness complexes," in *Eurographics Symposium on Point-Based Graphics*, ser. SPBG '04, M. Gross, H. Pfister, M. Alexa, and S. Rusinkiewicz, Eds. The Eurographics Association, 2004.

[37] O. Busaryev, T. K. Dey, and Y. Wang, "Tracking a generator by persistence," in *Computing and Combinatorics (COCOON)*, ser. Lecture Notes in Computer Science, vol. 6196. Berlin, Heidelberg: Springer Verlag, 2010, pp. 278–287.

[38] V. de Silva, D. Morozov, and M. Vejdemo-Johansson, "Dualities in persistent (co)homology," *Inverse Problems*, vol. 27, no. 12, 2011.

[39] M. ufar and iga Virk, "Fast computation of persistent homology representatives with involuted persistent homology," 2021.

[40] M. ufar, "Ripserer.jl: flexible and efficient persistent homology computation in julia," *Journal of Open Source Software*, vol. 5, no. 54, 2020.

[41] V. Satopa, J. Albrecht, D. Irwin, and B. Raghavan, "Finding a 'kneedle' in a haystack: Detecting knee points in system behavior," in *31st International Conference on Distributed Computing Systems Workshops*, ser. ICDCSW. IEEE Computer Society, 2011, pp. 166–171.

[42] N. O. Malott, A. Sens, and P. A. Wilsey, "Topology preserving data reduction for computing persistent homology," in *International Workshop on Big Data Reduction*, 2020.

[43] M. E. Muller, "A note on a method for generating points uniformly on n-dimensional spheres," *Communications of the ACM*, vol. 2, no. 4, pp. 19–20, Apr. 1959.