Earthline Journal of Mathematical Sciences
E-ISSN: 2581-8147
Volume 14, Number 2, 2024, Pages 349-378
https://doi.org/10.34198/ejms.14224.349378

Persistent Homology and Persistent Cohomology: A Review

Busayo Adeyege Okediji

Department of Pure and Applied Mathematics, Faculty of Pure and Applied Sciences,
Ladoke Akintola University of Technology, P.M.B. 4000, Ogbomoso, Nigeria
e-mail: oluwabusayo122833@gmail.com

Abstract

Persistent homology is an important tool in non-linear data reduction. Its sister theory, persistent cohomology, has attracted less attention in the past years eventhough it has many advantages. Several literatures dealing with theory and computations of persistent homology and cohomology were surveyed. Reasons why cohomology has been neglected over time are identified and, few possible solutions to the identified problems are made available. Furthermore, using Ripserer, the computation of persistent homology and cohomology using 2-sphere both manually and computationally are carried out. In both cases, same result was obtained, particularly in the computation of their barcodes which visibly revealed the point where the two coincides. Conclusively, it is observed that persistent cohomology is not only faster in computation than persistent homology, but also uses less memory in a little time.

1 Introduction

With the recent increase in the volume, the variety and the dimensionality of available data, then identifying, extracting, and exploiting their underlying

Received: October 31, 2023; Revised: December 28, 2023; Accepted: January 3, 2024; Published: January 8, 2024
2020 Mathematics Subject Classification: 55Nxx.
Keywords and phrases: persistent homology, persistent cohomology, Ripser.
*Corresponding author

structure have become an issue for fundamental importance for data analysis and statistical learning (Making sense of data). The good news is Topological Data Analysis (TDA) came as a viable field that profer viable solution to this problem. TDA is a recent and fast growing field providing a set of new topological and geometric tools to infer relevant features for possibly complex data. It proposes new well founded mathematical theories and computational tools that can be used independently or in combination with other data analysis and statistical learning techniques. Below is the pipeline of TDA.

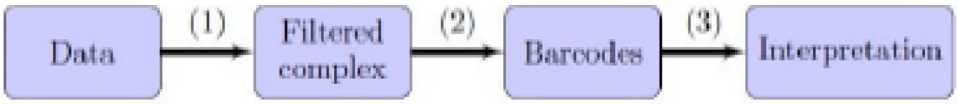

Figure 1: TDA Pipeline.
(Otter et al. 2017)

2 Some Applications of Persistent Homology and Persistent Cohomology

In this section, we provide a brief survey of the different domains in which topological data analysis, specifically persistent homology and persistent cohomology has found their applications.

2.1 General applications of persistent homology

Homology is a notion in Algebraic Topology associating a sequence of abelian groups or modules with a topological space. It can be very difficult to compute the homology of arbitrary topological spaces. Hence, spaces are approximated by combinatorial structures called SIMPLICIAL COMPLEXES for which homology can be computed algorithmically.

The idea of persistent homology has found application in several area of studies including robotics [1], sensor networks [3, 7], and the analysis of different other domains [8, 9, 13]. Several structures in material science has been analysed using persistent homology. Nakamura et al. [19] proposed a methodology based on persistent homology to describe Medium Range Order (MRO) in amorphous solids. These experiments on crystalline, random and amorphous structures showed the power of persistent diagrams to explain geometric structures in single and many-body atomic configurations. For instance in glass materials, the presence of characteristic curves in a function over the persistent diagram imply presence of MRO. For crystalline solids, the periodic structure yielded a few island supports in the persistence diagram with high multiplicity. The diagonal region in this diagram corresponds to the secondary holes that represent distortion from the primary holes. In fact, this paper provides strong evidence of the importance of persistent diagrams in the analysis of material structures. In another work, Hiraoka et al. [56] showed similar results where different states of matter where distinguished using persistent diagrams. Thermal flunctuations and strains on molecules were studied using persistent homology.

The authors of [23] identified the qualitative changes in time series using delay embeddings and topological data analysis. This delay-variant embedding method revealed multiple time - scale patterns in a time series. The authors also combined these features with the kernel technique in machine learning algorithms to classify the general time-series data. In another work, Perea and Harer [27] used a sliding window based technique and persistence to compute periodicity of signals.

In computer vision, persistent homology was used for a wide array of applications ranging from image segmentation [60] to shape characterization [6]. Carriere et al. [22] used persistent homology to estimate 3 dimensional (3D) shape. They worked on generating stable signatures to describe compact smooth surfaces in three dimensional euclidean space R^3. Given a shape S, persistence was calculated by growing geodesic balls for each point $x \in S$. The persistent diagrams were vectorised by treating them as metric spaces using pairwise distances. Thus for each shape, there exist a set of vectors, one for each point in the shape.

Multidimensional Scaling on these vectors showed that there are some continuity between vectors with identical labels, which suggests that the signatures vary continuously along the shape. It is these vectors that are in turn used as features for machine learning. Results of classifying shape based on these signatures are quite encouraging. In a similar application, Bonis et al. [10] used persistent homology to estimate 3D shape poses. This work used the intervals obtained in persistent homology as features for pooling in a bag-of-word approach. The result obtained was far better than the state-of-the-art techniques as well. In a work related to image analysis of medical data, Singh et al. [29] studied the cell arrangement of microscopic images of breast cancer using topological data analysis. They took the points and their corresponding weights as the individual nucleus and its mass. Using this, the persistent homology of the Vietoris Rips (VR) complex starting from a weighted point cloud was computed. Different cancer type was classified and demonstrated that the topological features contain useful complementary information to image - appearance based features that can improve discriminatory performance of classifiers. Recently, Leon et al. [30] used filters on gaits of human silhouettes to build simplicial complexes. They used these complexes for gender classification. In another work, Aras et al. [5] used persistent homology to analyse image tampering. They utilised the non uniformity in Local Binary Pattern of images to build simplicial complexes. The number of connected components in this simplicial complex were used for different threshold values as features for their classifier. The results obtained showed that the persistent homology sequence defines a discriminating criterion for the morphing attacks (i.e. distinguishing morphed images from genuine ones).

There has been experiments on different kernels using persistence including Sliced Wasserstein [21], persistence scale space [45], weighted Gaussian [62], and persistence sheer kernels on Riemannian Manifold [53]. Some works used persistent diagrams as features for machine learning classifiers by doing statistical measure on the intervals. The authors of [15, 17] used the binding process by collecting values at the grid point. Other works [16, 60] used the top 'n' intervals as features. Bubenik [12] introduced persistent landscapes which was

subsequently used as feature vectors either through binning or as intensity maps for neural networks. Recently the persistent diagrams has been represented using a persistent surface function [2]. The persistent surface can be discretized by a cartesian grid into image data. By the integration of persistent surface function over each grid (or pixel), a persistent image function was obtained. These images can in term be fed directly into classifiers.

In Bio-science, topological data analysis found applications in medical imaging [26, 31], molecular architecture [33, 50], and genomics [14], among others. Topological structures have also been used to analyse viruses. Emmett et al. worked on influenza [47] to show a bimodality of distribution of intervals in the persistent diagram. This bimodality indicated two scales of topological structure, corresponding to intra-subtype (involving one HA subtype) and inter-subtype (involving multiple HA subtypes) viral reassortments. These results on viruses suggested that persistent homology can be used to study other forms of reticulate evolution. Overall, this paper presented clear examples of topological structures demonstrating different biological processes. In another work, Parida et al. [34] used topological characteristics to detect subtle admixture in related populations. In [36], gene expressions from peripheral blood data was used to build a model based on TDA network model and discrete Morse theory to look into routes of disease progression. Persistent homology has also been employed in [42] for comparison of several weighted gene co-expression networks. Persistent Homology was used to identify DNA copy number aberrations [4]. Their experiments found a new amplification in 11q at the location of the progesterone receptor in the Luminal A subtype. Seemann et al. [43] used persistent homology to identify correlated patient samples in gene expression profiles. Their work focuses on the H_0 homology class which is used to partition the point cloud. The famous paper by Nicolau et al. [38] identified a subgroup of breast cancers using topological data analysis in gene expressions. Several works [40, 43] on use of machine learning techniques on gene expression profile have shown promising results. Kong et al. [61] used random forests to extract features for their Neural Network Architecture. They investigated a problem similar to our 'Topo-relevant gene' and the results

show significant improvement. [58] analyzes gene expression data to classify cancer types. Different techniques of supervised learning are used to understand genes and classify cancer. The authors of [44] used machine learning to identify novel diagnostic and prognostic markers and therapeutic targets for soft tissue sarcomas. This work unveiled the overlap of three groups of tumors in their molecular profile. The authors of [16] characterised proteins using persistent homology. The work used amino acid molecules as points to build a Vietoris Rips (VR) filtration. Based on this filtration, the authors chose a feature vector of length 13, based on number, length, birth and death time, average, and summation of certain specific intervals. This feature vector was used as input to supervise classification. This work showed that the classification accuracy for protein structures was improved even by choosing feature vectors naively without any statistical basis.

In another work [18], the authors used barcodes to describe the secondary structure of proteins such as alpha helix and beta sheets. They also employed the said barcodes to analyse protein elastic network models. Several works [40, 43] on the use of machine learning techniques on gene expression profile have shown promising results. Kong et al. [61] used random forests to extract features for their Neural Network Architecture. A problem similar to 'Topo-relevant gene' was investigated and the results showed significant improvement. [58] analyzed gene expression data to classify cancer types. Different techniques of supervised learning were used to understand genes and classify cancer. The authors of [44] used machine learning to identify novel diagnostic and prognostic markers and therapeutic targets for soft tissue sarcomas. This work revealed the overlap of three groups of tumors in their molecular profile.

The computation of representative cycles for homology groups with $\mathbb{Z}_2$ has been extensively studied over the decades While a polynomial time algorithm for computing an optimal basis for the first homology group H_1 [39] was proposed, finding an optimal basis for dimension greater than one and localizing a homology class of any dimension were proved NP-hard [25]. There are few works addressing the problem of finding representatives for persistent homology, some of which computed an optimal cycle at the birth index of an interval but

do not consider what actually died at the death index [48, 50]. Obayashi [46] formalizes the computation of optimal representatives for a finite interval as an integer programming problem. He advocated solving it with linear programs, though the correctness is not necessarily guaranteed. Wu et al. [51] proposed an algorithm for computing an optimal representative for a finite interval with a worst-case complexity exponential to the cardinality of the persistence diagram. Chambers et al.[23] proved that the localization problem over dimension one is NP-hard when the given simplicial complex is a 2-manifold. Several other works [11, 24, 35, 49] addressed variants of the two problems while considering special input classes, alternative cycle measures, or coefficients for homology other than $\mathbb{Z}_2$.

2.2 General applications of persistent cohomology

Cohomology is a sequence of Abelian groups associated to topological spaces and is defined from a cochain complex. The idea of persistent cohomology has found less applications in several area of studies. A lot of researchers prefered persistent homology over persistent cohomology. Counting less on this, cohomology has been applied in a certain little but effective capacity.

Chang and Guo [54] used persistent cohomology to extract data with multicomponent heterogeneous information where it was inferred that non geometric information can either be distributed globally or locally on the datasets in the geometric sense and can properly be defined on topological spaces. State Space discovery in the spatial representation system was evaluated using persistent cohomology [63]. [64] presented Javaplex as a research software package for computing persistent cohomology. It worked perfectly well until certain cohomological operations (cup products, and the morse function) surfaced. Many thanks to [55] who gave an improvement on this through the implementation of ripser. In 2011 as well, the authors of [55] gave the equivalence of persistent homology and persistent cohomology where duality was greatly executed. [65] in 2022 constructed spherical coordiantes from persistent cohomology. Authors

of [57] presented an approach to topological motion planning which is fully data driven in nature and which relies solely on the knowledge of samples in the free configuration space.

From the review above, it is obvious that persistent cohomology has gained less attention from researchers. Some of the presumed reasons for this are listed below.

• There is natural isomorphism that takes the persistent homology to persistent cohomology but there is no such canonical isomorphism that takes persistent cohomology to persistent homology.

• In terms of computations, the inadequate implementation of softwares which appropraitely takes care of the cohomological operations, and those which allows for clearing. However, the difference is not in the computation of their barcodes (which is the same for homology and cohomology), but in computing the (co)cycles which generate cohomology.

2.3 Suggested solution to the spotted problems

• By Universal Coefficient theorem, there is a natural isomorphism between homology and cohomology. (Theorem 3.8) Injecting this on the isomorphism of the barcodes, then the first problem proves solvable.

• Although there are many softwares (Plex, JavaPlex, PHAT, Dinoysus, ripser and others) which naturally compute homology in a little time, but takes countless hours before they could compute cohomology. The execution of Ripserer (a jula app) is a goodnews that gives a lasting solution to this problem, much more that it computes cohomology before homology in a little time with less memory, and also allows for clearing which speeds up the computation of persistence.

3 Persistent Theory

Persistence, as introduced by Edelsbrunner, Letscher, and Zomorodian [68] and refined by Zomorodian and Carlsson [66] is the major response to the problems of TDA. We shall give a brief overview of the theories, compute the persistent homology and persistent cohomology of 2-sphere using the application of these theories manually and computationally using Ripserer.

Definition 3.1. *[66] The chain complex is the sequence of chain groups connected by boundary homomorphism* $\delta_p : C_p \to C_{p+1}$ *such that* $C_{p-1} \circ C_p = 0$.

Definition 3.2. *[66] A p-chain c is a cycle if its boundary is empty, that is,* $\delta_c = 0$. *While a p-chain is a boundary if it is in the image of* δ_{p+1}, *that is,* $c = \delta d$, $p \in \mathbb{Z}$. *The p cycles form a group denoted by* $\mathbb{Z}_p$, *which is the kernel of the boundary homomorphism* δ_p. *The p boundaries also form a group denoted by* B_p *which is the image of the homomorphism.*

Since $\delta_p \circ \delta_{p+1}(c) = 0$, *for every integer p and every (p+1)-chain, then* $Im\delta_{p+1} = B_p \subseteq Ker\ \delta_p = \mathbb{Z}_p$. *That is, every p-boundary is also a p-cycle. Hence, the* ***pth homology group*** *of K is the quotient group* $H_p(K) = \frac{Z_p(k)}{B_p(k)} = \frac{Ker(\delta_p)}{Im(\delta_{p+1})}$ *containing equivalence classes of k-cycles.*

3.1 Persistent Homology [28, 66, 67]

Given a simplicial complex K, a **FILTRATION** of a SIMPLICIAL COMPLEX K is a nested collection of subcomplexes $k_1 \leq k_2 \leq k_3 \leq \cdots \leq k_m = k$, that is, a finite sequence of subcomplexes $K^f = K^f : 0 \leq a \leq m$ of K such that $\varnothing = K^0 \subseteq K^1 \subseteq K^2 \subseteq \cdots \subseteq K^m = K$ for $(a, b) \in 0, \cdots, m$ such that $a \leq b$. The (a, b)-**PERSISTENT** p**-homology** group $H_{\mathrm{p}}^{\mathrm{a,b}}(K)$ of simplicial complex K consists of k cycles included from $\zeta(K^a)$ into $\zeta(K^b)$ modulo boundaries. Formally, it can be defined as $H_{\mathrm{p}}^{\mathrm{a,b}}(K^f) = ImI_{\mathrm{p}}^{\mathrm{a,b}}$ where $I_{\mathrm{p}}^{\mathrm{a,b}}$ denotes the Linear map between $H_p(K^a)$ and $H_p(K^b)$ induced by the inclusion of complexes between K^a

and K^b.

Generally, while **HOMOLOGY** captures **CYCLES** in a shape by factoring out the boundary cycles, **PERSISTENT HOMOLOGY** allows for the retrieval of cycles that are non boundary elements in a certain step of the filtration and that will turn into **BOUNDARIES**. Also, homology define and Count **HOLES**, but persistent homology measure **HOLES.** Although there are several types of complexes, but for the purpose of this work, we shall restrict ourselves to FILTERED CELL COMPLEXES, since we are interested in the persistent topology of a filtered topological space which is a sequence X of cell complexes $X : X_1 \subset X_2 \subset \cdots X_n = X_\infty$ where X_1 is a vertex say (σ_1). Each complex is obtained from the previous one by adding a single cell: $X_i = X_{i-1} \cup \sigma_i$.

3.2 Cohomology Theory

Definition 3.3. *[55] A cochain complex is the sequence of chain groups connected by coboundary homomorphism* $\delta^p : C^p \to C^{p+1}$ *such that* $C^{p-1} \circ C^p = 0$.

Definition 3.4. *[55] A p-cochain c is a cocycle if its coboundary is empty, that is,* $\delta^c = 0$. *While a p-cochain is a coboundary if it is in the image of* δ^{p+1}, *that is,* $c = \delta d$, $p \in \mathbb{Z}$. *The p cocycles form a group denoted by* $\mathbb{Z}^p$, *which is the kernel of the coboundary homomorphism* δ^p. *The p coboundaries also form a group denoted by* B^p *which is the image of the homomorphism.*

Since $\delta^p \circ \delta^{p+1}(c) = 0$, *for every integer p and every* $(p+1)$ *chain, then* $Im\delta^{p+1} = B^p \subseteq coker\ \delta^p = \mathbb{Z}^p$. *That is, every p-coboundary is also a p-cocycle. Hence, the* ***pth cohomology group*** *of K is the quotient group* $H^p(K) = \frac{Z^p(k)}{B^p(k)} = \frac{coker(\delta^p)}{Im(\delta^{p+1})}$ *containing equivalence classes of k-cocycles.*

3.3 Persistent Cohomology [55, 66, 69]

From Definition 3.2, it could be inferred that the (a, b)-**PERSISTENT p-cohomology** group $H^{\mathrm{p}}_{\mathrm{a,b}}(K)$ of simplicial complex K consists of k cocycles included from $\zeta(K_a)$ into $\zeta(K_b)$ modulo coboundaries. Formally, it can be defined as $H^{\mathrm{p}}_{\mathrm{a,b}}(K^f) = \mathrm{Im}\ I^{\mathrm{p}}_{\mathrm{a,b}}$ where $I^{\mathrm{p}}_{\mathrm{a,b}}$ denotes the linear map between $H^p(K_a)$ and $H^p(K_b)$ induced by the inclusion of complexes between K_a and K_b.

3.4 Persistent Modules [13, 55]

An N-indexed persistence module over a field $\mathbb{F}$ is a sequence (v_0, a_0) of $\mathbb{F}$ vector spaces V_k and linear maps a_k defined for $k \geq 0$. The map (a_0 are not required to satisfy $(a_k \circ a_{k-1}) = 0$).

The four standard persistence modules (absolute homology, absolute cohomology, relative homology and relative cohomology) in relation to the filtered cell complexes are respectively given below.

(1) $H_*(X)\colon H_*(X_1) \to \cdots \to H_*(X_{n-1}) \to H_*(X_n)$

(2) $H^*(X)\colon H^*(X_1) \leftarrow \cdots \leftarrow H^*(X_{n-1}) \leftarrow H^*(X_n)$

(3) $H_*(X_\infty, X)\colon H_*(X_n) \to H_*(X_n, X_1) \to \cdots \to H_*(X_n, X_{n-1})$

(4) $H^*(X_\infty, X)\colon H^*(X_n) \leftarrow H^*(X_n, X_1) \leftarrow \cdots \leftarrow H^*(X_n, X_{n-1})$

Theorem 3.5. *[68] Every pointwise finite dimensional (P.F.D) persistence module $V \in Vect^{\mathbb{R}}$ decomposes uniquely (up to isomorphism) into interval persistence modules $C(I)$. $V \cong \oplus C(I)$ where $I \in B(V)$ where $B(V)$ is a multiset (that is, a set of objects with multiplicities) of intervals of the form $[a, b)$ or $(-\infty, b)$ for some $a \in \mathbb{R}, b \in \mathbb{R} \cup [+\infty]$. This $B(V)$ is called the* **BARCODE** *of V.*

To every barcode, there is always an associated persistent diagram.

Lemma 3.6. *[55, 65, 68] The persistence diagram (barcode) is the multi set of ordered pairs $[p, q]$ in the decomposition, or alternatively the multi set of half*

open intervals $[a_p, a_{q+1})$. *Thus we write* : ***Pers(H(X))***$= ([p_1, q_1], \cdots][p_m, q_m]$ $= ([a_{p1}, a_{q1+1}), \cdots [a_{pm}, a_{qm+1})$ *where* $1 \leq p \leq q \leq n$ *for (Absolute Homology and cohomology) and* $0 \leq p \leq q \leq n-1$ *for (Relative Homology and Cohomology) with the convention that* $a_0 = -\infty$ *and* $a_{n+1} = \infty$.

Lemma 3.7. *[55, 65, 68] Given a multi set of ordered pairs* $[p, q]$ *in any decomposition, the multi set of half open intervals* $[a_p, a_{q+1})$, *it is customary to discards points for which* $a_p = a_{q+1}$

Theorem 3.8. *[55]* $H^k(X; K) \cong$ *Hom* $(H_k(X; K), K)$. *There is a natural isomorphism between homology and cohomology.*

Proof. "Natural" implies that the induced maps $H_k(X_i; K) \to H_k(X_j; K)$ and $H^k(X_i; K) \to H^k(X_j; K)$ are adjoint and hence have the same rank. Because of the way the barcode is uniquely determined by dimensions and ranks, it follows that the absolute homology and cohomology barcodes are the same. This argument applies equally well to the relative barcodes. □

Theorem 3.9. *For all* $k, Pers(H_k(X)) = Pers(H^k(X))$: $Pers(H_k(X_\infty, X)) = Pers(H^k(X_\infty, X))$.

In other words, ***Homology and Cohomology have IDENTICAL BARCODES***.

3.5 Persistence diagram for Absolute homology and Cohomology for 2-sphere

For the cellular filtration of 2-sphere, there are six cells (say $(X_1, X_2, X_3, X_4, X_5$ and $X_6)$.

Writing this as a filtered cell complex, we have

$$X : X_1 \subset X_2 \subset X_3 \subset X_4 \subset X_5 \subset X_6 \cdots (1)$$

Now if we apply a homology functor $H(-)$ to equation (1) above, we have

$$H(X)\colon H(X_1) \subset H(X_2) \subset H(X_3) \subset H(X_4) \subset H(X_5) \subset H(X_6) \cdots (2)$$

where $H(-)$ denote the k - dimensional homology $H_k(-;k)$ or the total homology $H_*(-;k)$. Thus equation (2) above becomes a PERSISTENCE MODULE which decomposes as a direct sum of interval modules. These are labelled by the ordered pairs of integers $[p,q]$ where $1 \leq p \leq q \leq n$.

The pair $[p,q]$ indicates a feature which persists over the index set $(p, \cdots q)$ we frequently interpret $[p,q]$ as the half open real interval $[a_p, a_{q+1})$ with the convention that $a_{n+1} = \infty$, and so the persistence diagram or barcode for ABSOLUTE HOMOLOGY is written as :

$$Pers(H(X)) = ([p_1,q_1], \cdots [p_m,q_m]$$

which is equal to $([a_{p1}, a_{q+1}), \cdots [a_{pm}, a_{qm+1})$

Applying this to the cellular filtration of the 2-sphere, we thus have that $1 \leq p \leq q \leq 6$ and

$$Pers(H(X)) = [1,1],[1,2],[1,3],[1,4],[1,5],[1,6],$$

$$[2,2],[2,3],[2,4],[2,5],[2,6],[3,3],[3,4],[3,5],[3,6],$$

$$[4,4],[4,5],[4,6],[5,5],[5,6],[6,6] \cdots (3)$$

By Lemma 3.7, equation (3) thus becomes;

$$Pers(H(\mathbb{S})) = [1,6]_0,[2,2]_0,[4,4]_1,[6,6]_2 \cdots (4)$$

Lemma 3.6 thus make equation(4) becomes

$$PersH(\mathbb{S}) = [1,6+1)_0,[2,2+1)_0,[4,4+1)_1,[6,6+1)_2 \cdots (5)$$

Going by the convention that $a_{n+1} = \infty$, equation(5) thus becomes;

$$Pers(H(\mathbb{S}) = ([1,\infty)_0,[2,3)_0[4,5)_1,[6,\infty)_2)$$

And so $Pers(H_*(\mathbb{S}) = [1,6]_0, [2,2]_0, [4,4]_1, [6,6]_2$ which is equal to $([1,\infty)_0, [2,3)_0[4,5)_1, [6,\infty)_2$

The subscript k in $[p,q]_k$ or $[a_p, a_{q+1})_k$ indicates that the feature occurs in k-dimensional homology. However, the persistence diagram for absolute cohomology is a multiset of integer ordered pairs $[p,q]$ with $1 \leq p \leq q \leq n$.

3.6 Persistence diagram for the Relative Homology and Cohomology of 2-sphere

The persistence diagrams for relative homology and cohomology are multisets of pairs $[p,q]$ with $0 \leq p \leq q \leq n-1$. In all cases, we interpret $[p,q]$, as the half open interval $[a_p, a_{q+1})$, with the convention that $a_0 = -\infty$ and $a_{n+1} = \infty$. Unlike absolute homology and cohomology whose interval range from 1 to n, the interval for relative homology and cohomology ranges from 0 and $n-1$. Since we have six(6) cells for the cellular filtration of 2-sphere, then its interval for relative (homology and cohomology) range from 0 and 5.

The persistent module for relative homology and cohomology of 2-sphere thus becomes

$$Pers(H(X)) = [0,0], [0,1], [0,2], [0,3], [0,4], [0,5],$$
$$[1,1], [1,2], [1,3], [1,4], [1,5], [2,2], [2,3], [2,4],$$
$$[2,5][3,3], [3,4], [3,5], [4,4], [4,5], [5,5] \cdots (6)$$

Now applying Lemma 3.7 to equation (6), we have

$$Pers(H_*(S_6, \mathbb{S})) = [0,0]_0, [2,2]_1, [4,4]_2, [0,5]_2 \cdots (7)$$

And by Lemma 3.6, equation (7) becomes

$$Pers(H_*(S_6, \mathbb{S})) = [0, 0+1)_0, [2, 2+1)_1, [4, 4+1)_2, [0, 5+1)_2$$

So we have

$$Pers(H_*(S_6, \mathbb{S})) = [0,1)_0, [2,3)_1, [4,5)_2, [0,6)_2 \cdots (8)$$

And so going by the convention that $a_{n+1} = \infty$ and $a_0 = -\infty$, equation (8) thus becomes;

$$Pers(H_*(S_6, \mathbb{S})) = ([-\infty, 1)_0, [2, 3)_1 [4, 5)_2, [-\infty, 6)_2) \cdots (9)$$

For instance at index 2, we note that there is a non trivial element of $H_1(S_6, S_2)$ represented by any arc connecting the two points of S_2. To be specific, the homology class is $[\sigma_3] = [\sigma_4]$. This class varnishes in $H^1(S_6, S_3)$, and so it generates the interval $[2, 3)$ Looking vividly into (5) and (9), it is obvious that there is a close relationship between the barcodes for absolute homology and relative homology.

3.7 Computations using Ripserer

Ripserer is a pure Julia implementation of the Ripser algorithm for computations. Much like Ripser, it uses several computational tricks to achieve its speed. Among others, these include an implicit simplicial complex representation and the clearing optimization. In terms of performance, Ripserer is very close to Ripser, usually within around 30 percent. Ripserer's strength performance-wise is very sparse inputs, where it can sometimes outperforms Ripser. It also computes critical simplices that Ripser will naturally skip. It enable researchers to access state of algorithms for persistent homology, cohomology, hom complexes, filtered simplicial complexes, filtered cell complexes, witness complex constructions, and many more essential component of computational topology.

In terms of memory capacity and time usage, Ripserer is preferred over every other software packages, in that, Ripserer computes using 152MB within 1.2 seconds. This does not only show that Ripserer has an efficient memory over other software packages, but it also outperforms other codes by a factor of more than 40 in computation time, and a factor of more than 15 in memory efficiency. In short, it uses a little memory as possible and it is reasonable about computation time. To add to these, Ripserer has support for coefficients in a prime field.

The most interesting thing about Ripserer is that, it computes persistent cohomology by default. However it computes homology and cohomology using every form of complexes.

Our major aim is to obtain the persistence diagram(Barcodes) for both persistent homology and cohomology, and establish from the diagram that both have the same barcodes. The Ripserer codes for the computation of 2-Sphere generates the following diagrams.

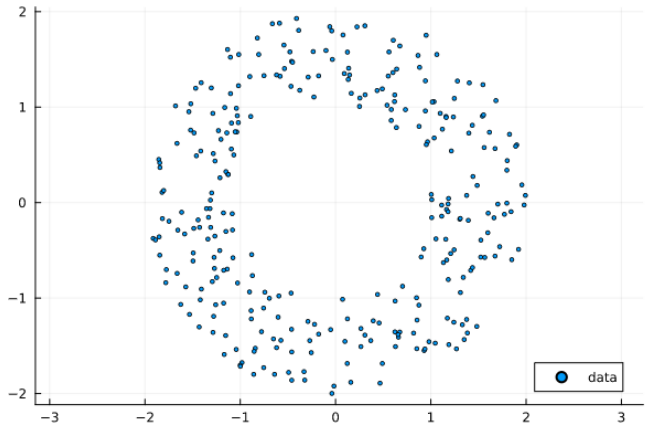

Figure 2: Scattered data.

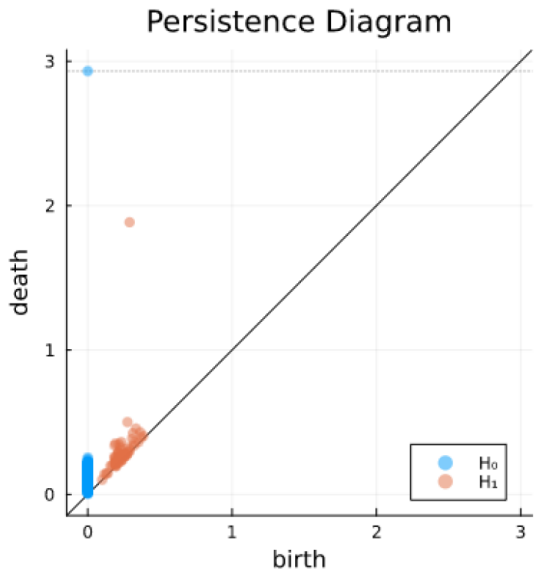

Figure 3: Persistence diagram for the data.

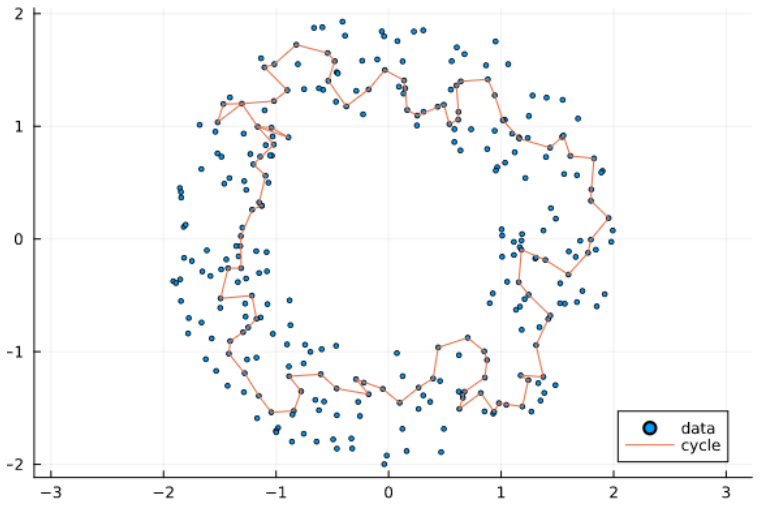

Figure 4: Persistence diagram for homology.

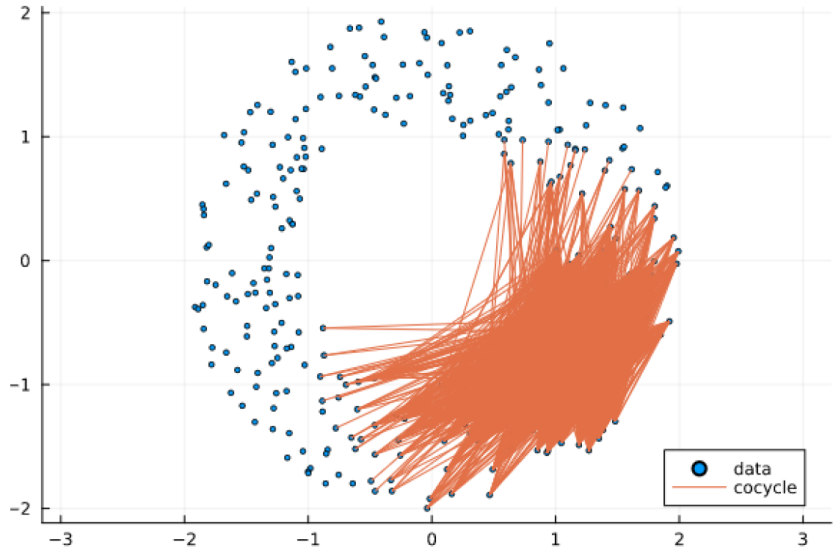

Figure 5: Persistence diagram for cohomology.

4 Conclusion

This work gave an eye opener to a probable solution to the persistent problem of the usage of cohomology. Over the years the usual way of computing cohomology was majorly to compute homology first (by reducing the generated boundary matrix to Smith Normal Form (SNF)), thereafter, obtain the cohomology (Which is the transpose of the boundary matrix (Coboundary matrix)). But through our review, it is clear that barcode isomorphism and universal coefficient theorem negate the claim, and the implementation of Ripserer which naturally computes cohomology before homology broke the archaic jinx, much more that persistent cohomology is not only faster in computation than persistent homology, but also uses less memory.

Appendix

RIPSERER CODES FOR GENERATING THE PERSISTENCE DIAGRAMS FOR HOMOLOGY AND COHOMOLOGY

- We start by loading some packages and generating some data.

```
using LinearAlgebra
using Plots
using Ripserer

function annulus(n, r1=1, r2=2, offset=(0, 0))
    result = Tuple{Float64,Float64}[]
    while length(result) < n
        point = 2 * r2 * rand(2) .- r2
        if r1 < norm(point) < r2
            push!(result, (point[1] + offset[1], point[2] + offset[2]))
        end
    end
    return result
end

data = annulus(300)

scatter(data; label="data", markersize=2, aspect_ratio=1)
```

Immediately after this command, there will be an output

- To obtain the persistence diagram, we give the command

```
diagram = ripserer(data)
plot(diagram)
```

The persistence diagram thus pop up as the outage. The diagram tells us that there is a persistent hole in the data, but tells us nothing about the location of the hole. Ripserer provides several methods to locate it. We will start with the simplest.

CRITICAL SIMPLICES

The first option is to find the death simplex of the interval. We start by extracting the interval in question. Keep in mind that the diagrams are sorted by persistence, so the last element will always be the most persistent. To obtain this, we give the command;

```
most_persistent = diagram[2][end]
```

There wil be an output too. In this output, it is noteworthy to emphasize that the interval has two simplices attached to it, the birth simplex and the death simplex. We can extract them with birth-simplex and death-simplex respectively. To obtain each of these, we give the command;

```
death_sx = death_simplex(most_persistent)
```

Since a simplex acts just like an array of indices, so it can be used to index into the data. To do this, we give the command;

```
data[death_sx]
```

• Ripserer also provides a Plots recipe for plotting simplices. It is invoked by passing the simplex and the data to plot. Not that only the edges of the simplices are plotted. To do this, we give the following command;

```
scatter(data; label="data", markersize=2, aspect_ratio=1)
plot!(death_sx, data; label="death simplex")
plot!(birth_simplex(diagram[2][end]), data; label="birth simplex")
```

After this command, we have a diagram as outage. The birth simplex is the simplex that first connects the hole. The death simplex is the simplex that fills the hole in, while the death simplex gives us a vague idea of where the hole is located.

• Ripserer also computes the representative cocycles. By default, Ripserer computes persistent cohomology. The resulting diagrams of persistent homology and cohomology are the same, but computing cohomology is much more efficient. When computing persistent cohomology, we can tell Ripserer to also compute representative cocycles. This is controlled with the reps keyword argument, using the command;

```
diagram_cocycles = ripserer(data; reps=true)
most_persistent_co = diagram_cocycles[2][end]
```

Regarding the output, notice that now, the interval also has a representative attached. The representative is an array of pairs with simplex value, where the value is the coefficient of the simplex. In reality, the type is different, but it acts exactly the same as a Pair. Giving the following command;

```
cocycle = representative(most_persistent_co)
```

We thus have our representative cocycles, which can be plotted in the same way as a simplex. To do this, we give the command;

```
scatter(data; label="data", markersize=2, aspect_ratio=1)
plot!(cocycle, data; label="cocycle")
```

Our output here is the persistence diagram of persistent cohomology. The cocycle is a collection of 1-simplices that, if removed, would break the cycle in our data set. This does not correspond to most people's intuitive understanding of a hole, but it can be useful in some situations. To find something more intuitive, we have to look to homology and its representative cycles.

• Ripserer can also compute the representative cycle (for homology). It supports two algorithms for computing representative cocycles. One is computing persistent homology directly, and the other is involuted homology computation. Involuted homology computes cohomology first and then uses its result to recompute cycles. While this increases the running time somewhat, it is still usually much more efficient than computing persistent homology directly. The difference is especially large for filtrations where the number of simplices increases quickly with dimension, such as Vietoris-Rips filtrations.

Involuted homology is computed by passing the argument alg=:involuted to Ripserer. If we wanted direct homology computation, we would use alg=:homology. The results for both cases are exactly the same. Note that invoking homology also turns on reps for dimensions one and higher. To do this, we give the following command;

```
diagram_cycles = ripserer(data; alg=:involuted)
most_persistent_ho = diagram_cycles[2][end]
```

If an interval with a representative is passed to plot, the representative is plotted. To do this, we give the following command;

```
scatter(data; label="data", markersize=2, aspect_ratio=1)
plot!(most_persistent_ho, data; label="cycle")
```

The output here gives the persistence diagram for persistent homology.

References

[1] Dirafzoon, A., & Lobaton, E. (2013). Topological mapping of unknown environments using an unlocalized robotic swarm. In *2013 IEEE/RSJ International Conference on Intelligent Robots and Systems, Tokyo, Japan*, 5545-5551. https://doi.org/10.1109/IROS.2013.6697160

[2] Adams, H., Emerson, T., Kirby, M., Neville, R., Peterson, C., Shipman, P., Chepushtanova, S., Hanson, E., Motta, F., & Ziegelmeier, L. (2017). Persistence images: A stable vector representation of persistent homology. *Journal of Machine Learning Research*, 18(8), 1-35.

[3] De Silva, V., & Ghrist, R. (2007). Coverage in sensor networks via persistent homology. *Algebraic and Geometric Topology*, 7(1), 339-358. https://doi.org/10.2140/agt.2007.7.339

[4] Arsuaga, J., Borrman, T., Cavalcante, R., Gonzalez, G., & Park, C. (2015). Identification of copy number aberrations in breast cancer subtypes using persistence topology. *Microarrays*, 4, 339-369. https://doi.org/10.3390/microarrays4030339

[5] Asaad, A., & Jassim, S. (2017). Topological data analysis for image tampering detection. In *Digital Forensics and Watermarking* (pp. 136-146). Lecture Notes in Computer Science, vol. 10431, Springer. https://doi.org/10.1007/978-3-319-64185-0_11

[6] Bendich, P., Edelsbrunner, H., & Kerber, M. (2010). Computing robustness and persistence for images. *IEEE Transactions on Visualization and Computer Graphics*, 16(6), 1251-1260. https://doi.org/10.1109/TVCG.2010.139

[7] Ghrist, R., & Muhammad, A. (2005). Coverage and hole-detection in sensor networks via homology. In *IPSN 2005. Fourth International Symposium on Information Processing in Sensor Networks*, 254-260. https://doi.org/10.1109/IPSN.2005.1440933

[8] Goldfarb, D. (2014). An application of topological data analysis to hockey analytics. arXiv preprint: arXiv:1409.7635.

[9] Li, M. Z., Ryerson, M. S., & Balakrishnan, H. (2019). Topological data analysis for aviation applications. *Transportation Research Part E: Logistics and Transportation Review*, 128, 149-174. https://doi.org/10.1016/j.tre.2019.05.017

[10] Bonis, T., Ovsjanikov, M., Oudot, S., & Chazal, F. (2016). Persistence based pooling for shape pose recognition. In *Computational Topology in Image Context* (pp. 19-29). Lecture Notes in Computer Science, vol. 9667, Springer. https://doi.org/10.1007/978-3-319-39441-1_3

[11] Borradaile, G., Chambers, E. W., Fox, K., & Nayyeriy, A. (2017). Minimum cycle and homology bases of surface embedded graphs. *Journal of Computational Geometry*, 8(2).

[12] Bubenik, P. (2015). Statistical topological data analysis using persistence landscapes. *Journal of Machine Learning Research*, 16(1), 77-102.

[13] Savic, A., Toth, G., & Duponchel, L. (2017). Topological data analysis (TDA) applied to reveal pedogenetic principles of European topsoil system. *Science of the Total Environment*, 586, 1091-1100. https://doi.org/10.1016/j.scitotenv.2017.02.095

[14] Camara, P. (2017). Topological methods for genomics: present and future directions. *Curr. Opin. Syst. Biol.*, 1, 95-101. https://doi.org/10.1016/j.coisb.2016.12.007

[15] Cang, Z., Mu, L., & Wei, G. (2018). Representability of algebraic topology for biomolecules in machine learning based scoring and virtual screening. *PLOS Computational Biology*, 14(1), e1005929. https://doi.org/10.1371/journal.pcbi.1005929

[16] Cang, Z., Mu, L., Wu, K., Opron, K., Xia, K., & Wei, G. (2015). A topological approach for protein classification. *Computational and Mathematical Biophysics*, 3(1). https://doi.org/10.1515/mlbmb-2015-0009

[17] Cang, Z., & Wei, G. (2018). Integration of element specific persistent homology and machine learning for protein-ligand binding affinity prediction. *International Journal for Numerical Methods in Biomedical Engineering*, 34(2). https://doi.org/10.1002/cnm.2914

[18] Cang, Z., & Wei, G. (2017). TopologyNet: Topology based deep convolutional and multi-task neural networks for biomolecular property predictions. *PLoS Computational Biology*, 13(7), e1005690. https://doi.org/10.1371/journal.pcbi.1005690

[19] Nakamura, T., Hiraoka, Y., Hirata, A., Escolar, E. G., & Nishiura, Y. (2015). Persistent homology and many-body atomic structure for medium range order in the glass. *Nanotechnology*, 26(30), 304001. https://doi.org/10.1088/0957-4484/26/30/304001

[20] Carlsson, G., Zomorodian, A., Collins, A., & Guibas, L. (2004). Persistence barcodes for shapes. In *SGP '04: Proceedings of the 2004 Eurographics/ACM SIGGRAPH Symposium on Geometry Processing* (pp. 124-135). ACM. https://doi.org/10.1145/1057432.1057449

[21] Carriere, M., Oudot, S., & Ovsjanikov, M. (2017). Sliced Wasserstein Kernel for Persistence Diagrams. In *ICML 2017 - Thirty-fourth International Conference on Machine Learning* (pp. 1-10). Sydney, Australia.

[22] Carriere, M., Oudot, S. Y., & Ovsjanikov, M. (2015). Stable topological signatures for points on 3D shapes. In *Proceedings of the Eurographics Symposium on Geometry Processing, SGP'15* (pp. 1-12). Aire-la-Ville, Switzerland: Eurographics Association. https://doi.org/10.1111/cgf.12692

[23] Tran, Q. H., & Hasegawa, Y. (2019). Topological time-series analysis with delay-variant embedding. *Phys. Rev. E*, 99(3), 032209. https://doi.org/10.1103/PhysRevE.99.032209

[24] Chen, C., & Freedman, D. (2010). Measuring and computing natural generators for homology groups. *Computational Geometry*, 43(2), 169-181. https://doi.org/10.1016/j.comgeo.2009.06.004

[25] Chen, C., & Freedman, D. (2011). Hardness results for homology localization. *Discrete and Computational Geometry*, 45(3), 425-448. https://doi.org/10.1007/s00454-010-9322-8

[26] Chung, M., Bubenik, P., & Kim, P. (2009). Persistence diagrams of cortical surface data. In *Information processing in medical imaging* (pp. 386-397). Springer. https://doi.org/10.1007/978-3-642-02498-6_32

[27] Perea, J. A., & Harer, J. (2015). Sliding windows and persistence: An application of topological methods to signal analysis. *Foundations of Computational Mathematics*, 15(3), 799-838. https://doi.org/10.1007/s10208-014-9206-z

[28] Kwitt, R., Huber, S., Niethammer, M., Lin, W., & Bauer, U. (2015). Statistical topological data analysis - a kernel perspective. In *Advances in Neural Information Processing Systems* 28 (pp. 3070-3078). Curran Associates, Inc.

[29] Singh, N., Couture, H. D., Marron, J. S., Perou, C., & Niethammer, M. (2014). Topological descriptors of histology images. In *Machine Learning in Medical Imaging* (pp. 231-239). Cham: Springer International Publishing. https://doi.org/10.1007/978-3-319-10581-9_29

[30] Leon, J. L., Cerri, A., Reyes, E. G., & Diaz, R. G. (2013). Gait-based gender classification using persistent homology. In *Progress in Pattern Recognition, Image Analysis, Computer Vision, and Applications* (pp. 366-373). Berlin, Heidelberg: Springer Berlin Heidelberg. https://doi.org/10.1007/978-3-642-41827-3_46

[31] Turner, K., Mukherjee, S., & Boyer, D. M. (2014). Persistent homology transform for modeling shapes and surfaces. *Information and Inference: A Journal of the IMA*, 3(4), 310â344. https://doi.org/10.1093/imaiai/iau011

[32] Pike, J. A., Khan, A. O., Pallini, C., Thomas, S. G., Mund, M., Ries, J., Poulter, N. S., Styles, I. B. (2020). Topological data analysis quantifies biological nano-structure from single molecule localization microscopy. *Bioinformatics (Oxford, England)*, 36(5), 1614-1621. https://doi.org/10.1093/bioinformatics/btz788

[33] Sauerwald, N., Shen, Y., & Kingsford, C. (2019). Topological data analysis reveals principles of chromosome structure through cellular differentiation. bioRxiv 540716. https://doi.org/10.1101/540716

[34] Parida, L., Utro, F., Yorukoglu, D., Carrieri, A. P., Kuhn, D., & Basu, S. (2015). Topological signatures for population admixture. In T. M. Przytycka (Ed.), *Research in Computational Molecular Biology* (pp. 261-275). Cham: Springer International Publishing. https://doi.org/10.1007/978-3-319-16706-0_27

[35] Dey, T. K., Hirani, A. N., & Krishnamoorthy, B. (2011). Optimal homologous cycles, total unimodularity, and linear programming. *SIAM Journal on Computing*, 40(4), 1026-1044. https://doi.org/10.1137/100800245

[36] Schofield, J. P. R., Strazzeri, F., Bigler, J., et al. (2019). A topological data analysis network model of asthma based on blood gene expression profiles. bioRxiv. https://doi.org/10.1101/516328

[37] Wheelock, K., Sun, K., Pandis, I., et al. (2019). A topological data analysis network model of asthma based on blood gene expression profiles. bioRxiv

[38] Nicolau, M., Levine, A. J., & Carlsson, G. (2011). 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*, 108(17), 7265-7270. https://doi.org/10.1073/pnas.1102826108

[39] Dey, T. K., Sun, J., & Wang, Y. (2010). Approximating loops in a shortest homology basis from point data. In *Proceedings of the Twenty-sixth Annual Symposium on Computational Geometry* (pp. 166-175). ACM.

[40] Pirooznia, M., Yang, J. Y., Yang, M. Q., & Deng, Y. (2008). A comparative study of different machine learning methods on microarray gene expression data. *BMC Genomics*, 9 (Suppl 1), S13. https://doi.org/10.1186/1471-2164-9-S1-S13

[41] Le, T., & Yamada, M. (2018). Persistence Fisher kernel: A Riemannian manifold kernel for persistence diagrams. In *Advances in Neural Information Processing Systems* 31 (pp. 10007-10018). Curran Associates, Inc.

[42] Duman, A. N., & Pirim, H. (2018). Gene coexpression network comparison via persistent homology. *International Journal of Genomics*, 11. https://doi.org/10.1155/2018/7329576

[43] Tarek, S., Abd Elwahab, R., & Shoman, M. (2017). Gene expression-based cancer classification. *Egyptian Informatics Journal*, 18(3), 151-159. https://doi.org/10.1016/j.eij.2016.12.001

[44] van IJzendoorn, D. G. P., Szuhai, K., Briaire-de Bruijn, I. H., Kostine, M., Kuijjer, M. L., & Bovee, J. V. M. G. (2019). Machine Learning Analysis of gene expression data reveals novel diagnostic and prognostic biomarkers and identifies 101 therapeutic targets for soft tissue sarcomas. *PLoS Computational Biology*, 15(2), e1006826. https://doi.org/10.1371/journal.pcbi.1006826

[45] Reininghaus, J., Huber, S., Bauer, U., & Kwitt, R. (2015). A stable multi-scale kernel for topological machine learning. In *2015 IEEE Conference on Computer*

Vision and Pattern Recognition (CVPR) (pp. 4741-4748). https://doi.org/10.1109/CVPR.2015.7299106

[46] Obayashi, I. (2017). Volume optimal cycle: Tightest representative cycle of a generator on persistent homology. arXiv preprint arXiv:1712.05103.

[47] Emmett, K., Rosenbloom, D., Camara, P., & Rabadan, R. (2014). Parametric inference using persistence diagrams: A case study in population genetics. arXiv.

[48] Emmett, K., Schweinhart, B., & Rabadan, R. (2016). Multiscale topology of chromatin folding. In *Proceedings of the 9th EAI International Conference on Bio-inspired Information and Communications Technologies* (pp. 177-180). https://doi.org/10.4108/eai.3-12-2015.2262453

[49] Erickson, J., & Whittlesey, K. (2005). Greedy optimal homotopy and homology generators. In *Proceedings of the Sixteenth Annual ACM-SIAM Symposium on Discrete Algorithms* (pp. 1038-1046).

[50] Escolar, E. G., & Hiraoka, Y. (2016). Optimal cycles for persistent homology via linear programming. In *Optimization in the Real World* (pp. 79-96). Springer. https://doi.org/10.1007/978-4-431-55420-2_5

[51] Wu, P., et al. (2017). Optimal topological cycles and their application in cardiac trabeculae restoration. In M. Niethammer et al. (Eds.), *Information Processing in Medical Imaging. IPMI 2017* (Lecture Notes in Computer Science, Vol. 10265). Springer, Cham. https://doi.org/10.1007/978-3-319-59050-9_7

[52] Krizhevsky, A. (2009). Learning Multiple Layers of Features from Tiny Images. April 8, 2009. https://www.cs.toronto.edu/~kriz/learning-features-2009-TR.pdf

[53] Mandal, S., Guzman-Saenz, A., Haiminen, N., Basu, S., & Parida, L. (2020). A topological data analysis approach on predicting phenotypes from gene expression data. In C. Martin-Vide, M. Vega-Rodriguez, & T. Wheeler (Eds.), *Algorithms for Computational Biology. AlCoB 2020* (Lecture Notes in Computer Science, Vol. 12099). Springer, Cham. https://doi.org/10.1007/978-3-030-42266-0_14

[54] Chang, Z., & Wei-Wei, G. (2020). Persistent cohomology for data with multicomponent heterogeneous information. *SIAM J Math Data Sci.*, 2(2), 396-418. https://doi.org/10.1137/19M1272226

[55] De Silva, D., Morozov, D., & Vejdemo-Johansson, M. (2011). Dualities in persistent (co)homology. *Inverse Problems*, 27(121), 124003. https://doi.org/10.1088/0266-5611/27/12/124003

[56] Hiraoka, Y., Nakamura, T., Hirata, A., Escolar, E. G., Matsue, K., & Nishiura, Y. (2016). Hierarchical structures of amorphous solids characterized by persistent homology. *Proceedings of the National Academy of Sciences*, 113(26), 7035-7040. https://doi.org/10.1073/pnas.1520877113

[57] Pokorny, F., Kjellstrom, H., Kragic, D., & Ek, C. (2012). Persistent Homology for Learning Densities with Bounded Support. In F. Pereira, C. J. Burges, L. Bottou, & K. Q. Weinberger (Eds.), *Advances in Neural Information Processing Systems* (Vol. 25). Curran Associates, Inc.

[58] Hwang, K. B., Cho, D. Y., Park, S. W., Kim, S. D., & Zhang, B. T. (2002). Applying machine learning techniques to analysis of gene expression data: cancer diagnosis. In S.M. Lin & K.F. Johnson (Eds.), *Methods of Microarray Data Analysis*. Springer, Boston, MA. https://doi.org/10.1007/978-1-4615-0873-1_13

[59] Liang, J., Edelsbrunner, H., Fu, P., Sudhakar, P. V., & Subramaniam, S. (1998). Analytical shape computation of macromolecules: II. molecular area and volume through alpha shape. *Proteins*, 33(1), 18-29. https://doi.org/10.1002/(SICI)1097-0134(19981001)33:1<18::AID-PROT2>3.0.CO;2-H

[60] Kurlin, V. (2015). A fast persistence-based segmentation of noisy 2D clouds with provable guarantees. *Pattern Recognition Letters*, 83, 3-12. https://doi.org/10.1016/j.patrec.2015.11.025

[61] Kong, Y., & Yu, T. (2018). A deep neural network model using random forest to extract feature representation for gene expression data classification. *Scientists Reports*, 8(1), 16477. https://doi.org/10.1038/s41598-018-34833-6

[62] Kusano, G., Hiraoka, Y., & Fukumizu, K. (2016). Persistence weighted Gaussian kernel for topological data analysis. In *Proceedings of The 33rd International Conference on Machine Learning*, 48, 2004-2013. Available from https://proceedings.mlr.press/v48/kusano16.html

[63] Kang, L., Xu, B., & Morozov, D. (2021). Evaluating state space discovery by persistent cohomology in the spatial representation system. *Frontiers in Computational Neuroscience*, 15. https://doi.org/10.3389/fncom.2021.616748

[64] Adams, H., Tausz, A., Vejdemo-Johansson, M. (2014). javaPlex: A research software package for persistent (co)homology. In *Mathematical Software, ICMS 2014 - 4th International Congress, Proceedings* (pp. 129-136). (Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics); Vol. 8592 LNCS). Springer Verlag. https://doi.org/10.1007/978-3-662-44199-2_23

[65] Schonsheck, N. C., & Schonsheck, S. C. Spherical coordinates from persistent cohomology. arXiv.org/pdf/2209.02791

[66] Zomorodian, A., & Carlsson, G. (2005). Computing Persistent Homology. *Discrete Comput. Geom.*, 33, 249-274. https://doi.org/10.1007/s00454-004-1146-y

[67] Xia, K., & Wei, G. W. (2014). Persistent Homology analysis of protein structure, flexibility, and folding. *IJNMBE*, 30(8), 814-844. https://doi.org/10.1002/cnm.2655

[68] Edelsbrunner, H., Letscher, D., & Zomorodian, A. (2002). Topological persistence and simplification. *Discrete Comput. Geom.*, 28(4), 511-533. https://doi.org/10.1007/s00454-002-2885-2

[69] Lowe, D. G. (1999). Object recognition from local scale-invariant features. *Proceedings of the Seventh IEEE International Conference on Computer Vision*, 2, 1150-1157. https://doi.org/10.1109/ICCV.1999.790410