



Field Choice Problem in Persistent Homology

Ippei Obayashi^{1,2} · Michio Yoshiwaki^{2,3}

Received: 19 November 2020 / Revised: 11 February 2023 / Accepted: 26 February 2023 /
Published online: 18 August 2023
© The Author(s) 2023

Abstract

This paper tackles the problem of coefficient field choice in persistent homology. When we compute a persistence diagram, we need to select a coefficient field before computation. We should understand the dependence of the diagram on the coefficient field to facilitate computation and interpretation of the diagram. We clarify that the dependence is strongly related to the torsion part of \mathbb{Z} relative homology in the filtration. We show the sufficient and necessary conditions of the independence of coefficient field choice. An efficient algorithm is proposed to verify the independence. A slight modification of the standard persistence algorithm gives the verification algorithm. In a numerical experiment with the algorithm, a persistence diagram rarely changes even when the coefficient field changes if we consider a filtration in \mathbb{R}^3 . The experiment suggests that, in practical terms, changes in the field coefficient will not change persistence diagrams when the data are in \mathbb{R}^3 .

Keywords Topological data analysis · Persistent homology · Algorithm · Algebraic topology

Mathematics Subject Classification 55U99 · 55N35

Editor in Charge: Kenneth Clarkson

Ippei Obayashi
i.obayashi@okayama-u.ac.jp

Michio Yoshiwaki
michio.yoshiwaki@a.riken.jp; michio.yoshiwaki@omu.ac.jp; yoshiwaki@sci.osaka-cu.ac.jp

- ¹ Center for Artificial Intelligence and Mathematical Data Science, Okayama University, 3-1-1 Tsushima-naka, Kita-ku, Okayama 700-8530, Japan
- ² Center for Advanced Intelligence Project, RIKEN, Nihonbashi 1-chome Mitsui Building, 15th Floor, 1-4-1 Nihonbashi, Chuo-ku, Tokyo 103-0027, Japan
- ³ Present Address: Osaka Central Advanced Mathematical Institute, 3-3-138 Sugimoto, Sumiyoshi-ku, Osaka 558-8585, Japan

1 Introduction

Topological data analysis (TDA) [4, 13] is, as the name suggests, the application of topology to data analysis. Persistent homology [14, 33] is one of the most important tools for TDA. In persistent homology, by encoding information on length scales in filtrations, we can capture characteristic geometric features with multiple length scales. By using filtrations, persistent homology is also robust to noise [7, 9]. Homology itself is translation and rotation invariant, and so persistent homology is similarly invariant. These properties are suitable for the analysis of shapes of data, and persistent homology is applied in various practical data analysis contexts in domains such as biology [6], image processing [20], and materials science [18, 21, 23, 30].

To describe our problem, we first define persistent homology. Persistent homology is defined on a filtration, an increasing sequence of topological spaces. We consider the following filtration:

$$\mathbb{X}: \emptyset = X_0 \subset X_1 \subset \dots \subset X_N.$$

The q th persistent homology $H_q(\mathbb{X}; \mathbb{k})$ with a coefficient ring \mathbb{k} is defined as follows:

$$H_q(\mathbb{X}; \mathbb{k}): H_q(X_0; \mathbb{k}) \rightarrow H_q(X_1; \mathbb{k}) \rightarrow \dots \rightarrow H_q(X_N; \mathbb{k}),$$

where $H_q(X_i; \mathbb{k}) \rightarrow H_q(X_{i+1}; \mathbb{k})$ is the homology map induced by the inclusion map $X_i \hookrightarrow X_{i+1}$. For any $i < j$, ϕ_i^j denotes the map from $H_q(X_i; \mathbb{k})$ to $H_q(X_j; \mathbb{k})$ induced by the inclusion map $X_i \hookrightarrow X_j$. By definition, the map ϕ_i^j coincides with the composite $\phi_{j-1}^j \circ \dots \circ \phi_i^{i+1}$ of the consecutive maps $\phi_{j-1}^j, \dots, \phi_i^{i+1}$.

When \mathbb{k} is a field, $H_q(\mathbb{X}; \mathbb{k})$ is known to have a good structure called interval decomposition. To explain the interval decomposition, we define interval indecomposables.

Definition 1.1 For $b \in \{1, \dots, N\}$ and $d \in \{1, \dots, N, \infty\}$ satisfying $b < d$, an interval indecomposable $I(b, d)$ is the following $N + 1$ vector spaces connected by N linear maps:

$$I(b, d) = 0 \rightarrow \dots \rightarrow 0 \rightarrow \underbrace{\mathbb{k}}_{b\text{th}} \rightarrow \dots \rightarrow \underbrace{\mathbb{k}}_{(d-1)\text{th}} \rightarrow \underbrace{0}_{d\text{th}} \rightarrow \dots \rightarrow 0, \quad \text{if } d \neq \infty,$$

$$I(b, d) = 0 \rightarrow \dots \rightarrow 0 \rightarrow \underbrace{\mathbb{k}}_{b\text{th}} \rightarrow \dots \rightarrow \mathbb{k}, \quad \text{if } d = \infty,$$

where $\mathbb{k} \rightarrow \mathbb{k}$ is an identity map and $0 \rightarrow \mathbb{k}, \mathbb{k} \rightarrow 0, 0 \rightarrow 0$ are zero maps.

Using interval indecomposables, we describe the following structure theorem of persistent homology.

Theorem 1.2 *For a filtration \mathbb{X} , the q th persistent homology $H_q(\mathbb{X}; \mathbb{k})$ has the following unique decomposition if \mathbb{k} is a field and $\dim H_q(X_i; \mathbb{k}) < +\infty$ for all i :*

$$H_q(\mathbb{X}; \mathbb{k}) \simeq \bigoplus_{m=1}^L I(b_m, d_m),$$

where $b_m \in \{1, \dots, N\}$, $d_m \in \{1, \dots, N, \infty\}$, and $b_m < d_m$.

This theorem depends on the fact that the polynomial ring $\mathbb{k}[z]$ with a field \mathbb{k} is a PID [33], and so this theorem does not hold for $\mathbb{k} = \mathbb{Z}$. Therefore, we use a field as a homology coefficient field even though standard homology theory often uses \mathbb{Z} as a coefficient ring.

When this interval decomposition is given, we define the q th persistence diagram (PD) $D_q(\mathbb{X}; \mathbb{k})$ as the multiset of pairs of endpoints of the interval indecomposables. That is, $D_q(\mathbb{X}; \mathbb{k}) = \{(b_m, d_m)\}_{i=1}^L$. Each pair is called a *birth-death pair*. b_m and d_m are called *birth time* and *death time* of the birth-death pair (b_m, d_m) , and $d_m - b_m$ is called a *lifetime* of the pair. Since a birth-death pair with a long lifetime corresponds to a “stable” homological structure in the filtration, we can use lifetimes to compare the significance of birth-death pairs.

Normally, we choose \mathbb{k} as one of \mathbb{R} , \mathbb{Q} , and $\mathbb{Z}_p = \mathbb{Z}/p\mathbb{Z}$ for a prime p . \mathbb{Z}_2 is most often used since it is amenable to a fast algorithm and an intuitive interpretation. Even if any k gives the same PD, we may not know a priori that this is the case, and there is still an algorithmic problem (whose solution is a main contribution of the paper). However, this is not practical because the dimensions of homology vector spaces for the same topological space are different when the \mathbb{Z} -homology group of the space has a non-zero torsion subgroup. For example, if a Klein bottle appears in a filtration, the PDs for $\mathbb{k} = \mathbb{Z}_2$ and $\mathbb{k} = \mathbb{R}$ are different since the 1st homology group of a Klein bottle is isomorphic to $\mathbb{Z} \oplus \mathbb{Z}_2$ and therefore $\dim H_1((\text{Klein bottle}); \mathbb{R}) = 1$ and $\dim H_1((\text{Klein bottle}); \mathbb{Z}_2) = 2$.

Throughout this paper, we often refer to a non-trivial torsion subgroup of relative homology groups $H_q(X_i, X_j; \mathbb{Z})$ in a filtration \mathbb{X} as torsion in the filtration.

In a previous study, a non-zero torsion subgroup has been a practical problem. Carlsson et al. [5] analyzed the space of small (3×3) image patches collected from natural images. In that study, a Klein bottle, a topological structure with a non-trivial torsion subgroup in its homology subgroup, played an important role. Those authors used persistent homology with \mathbb{Z}_2 coefficient field to analyze the data so it was difficult to directly find evidence of the Klein bottle. Therefore, they verify the Klein bottle by using \mathbb{Z}_2 -persistent homology and \mathbb{Z}_3 -persistent homology. Kahle et al. [22] also investigated a torsion subgroup in numerical experiments. They showed that a randomly generated simplicial d -subcomplex in $(n - 1)$ -simplex often has a non-trivial torsion subgroup for relatively large n (for example, $n = 75$). Therefore, the following questions naturally arise.

- What condition ensures the independence of the choice of the field \mathbb{k} ?

- If there is such a condition, is there an efficient algorithm to check it?
- If the algorithm shows the dependence of the choice of the field \mathbb{k} , how can we compute persistence diagrams for multiple coefficient fields?
- How often does $D_q(\mathbb{X}; \mathbb{k})$ change as the field \mathbb{k} changes?
- When $D_q(\mathbb{X}; \mathbb{k})$ changes depending on the choice of \mathbb{k} , how does $D_q(\mathbb{X}; \mathbb{k})$ change?

In this paper, we offer complete answers for the first and second questions and partial answers for the fourth and fifth questions. For the third question, some previous results are available to answer.

We remark that \mathbb{R} -parametrized filtration is also used for persistent homology. $\{X_t\}_{t \in \mathbb{R}}$ is \mathbb{R} -parametrized filtration if X_t is a topological space for each t and $X_t \subset X_s$ if $t < s$. Under a certain finiteness condition, \mathbb{R} -parametrized filtration can be regarded as an \mathbb{N} -parametrized filtration $X_0 \subset \dots \subset X_N$. Therefore \mathbb{N} -parametrized filtrations are used except for Theorem 1.17.

1.1 Known Algorithmic Approaches

Two algorithmic approaches to our problem are already known. One is the modular reconstruction algorithm and the other is the omni-field persistence algorithm. Both algorithms compute persistence diagrams for multiple coefficient fields simultaneously. These algorithms are based on the standard PH algorithm [14, 27, 33] shown in Algorithm 2.1.

An algorithm proposed by Maria and Boissonnat [2], called the modular reconstruction algorithm, is available to compute persistence diagrams for multiple coefficient fields efficiently. The algorithm computes diagrams for multiple fields $\mathbb{Z}_{q_1}, \dots, \mathbb{Z}_{q_r}$ simultaneously using the isomorphism $\mathbb{Z}_{q_1} \times \dots \times \mathbb{Z}_{q_r} \simeq \mathbb{Z}/(q_1 \dots q_r)\mathbb{Z}$, which is the well-known Chinese remainder theorem. The information of all diagrams is recorded in a single $[\mathbb{Z}/(q_1 \dots q_r)\mathbb{Z}]$ -matrix and left-to-right reductions are applied to the matrix. We can extract the diagrams $D_q(\mathbb{X}; \mathbb{Z}_{q_1}), \dots, D_q(\mathbb{X}; \mathbb{Z}_{q_r})$ from the matrix. The time complexity of the modular reconstruction algorithm in the worst case is the same as the persistent homology algorithm in big O notation. Still, the constant factor in the time complexity of the modular reconstruction algorithm is larger since the algorithm records more information and needs the operations taking modulo more.

Omni-field persistence¹ in Dionysus2² by Morozov is another option for computing persistence diagrams for multiple coefficient fields. Omni-field persistence basically uses a matrix with coefficients in \mathbb{Q} to record the information of diagrams. Each time the algorithm encounters a division by a prime number p , a separate \mathbb{Z}_p -matrix is prepared to keep the information about the \mathbb{Z}_p persistent homology, and the information of all the rest primes is kept in the \mathbb{Q} -matrix. Left-to-right reductions are applied to all matrices separately. Any paper about omni-field persistence is not published yet, but the algorithm probably works as intended. Omni-field persistence is slower

¹ <https://mrzv.org/software/dionysus2/tutorial/omni-field.html>.

² <https://mrzv.org/software/dionysus2/index.html>.

than the modular reconstruction algorithm since omni-field persistence keeps separate matrices. However, on the other hand, omni-field persistence computes \mathbb{Z}_p -persistence diagrams for all prime numbers p . The information is richer than the modular reconstruction algorithm.

1.2 Results

To describe the results of the paper, we give some assumptions. We always assume the finiteness of the filtration. A filtration is *finite* if X_N is a finite simplicial/cell/cubical complex. This condition ensures the existence and uniqueness of the interval decomposition.

Question 1.3 When is $D_q(\mathbb{X}; \mathbb{k})$ independent of the choice of the field \mathbb{k} ?

To consider Question 1.3, it is desirable for the following conjecture to hold since $H_q(X_n; \mathbb{Z})$ is often free for every n in practical cases.

Conjecture 1.4 (incorrect!) *If $H_q(X_n; \mathbb{Z})$ is free for every n , then the persistent homology $H_q(\mathbb{X}; \mathbb{k})$ has the same decomposition for any field \mathbb{k} .*

However, we have a counterexample of this proposition (Fig. 1), which has been considered in the topological time series analysis literature [28, Sect. 3]. The authors of the paper constructed an example of the persistence diagram defined by a Möbius band and they verified that the persistence diagram is dependent on the field of coefficients.

Example 1.5 Let M be a Möbius strip and ∂M be its boundary. Both $H_1(\partial M; \mathbb{Z})$ and $H_1(M; \mathbb{Z})$ are isomorphic to \mathbb{Z} , and the homomorphism

$$H_1(\partial M; \mathbb{Z}) \rightarrow H_1(M; \mathbb{Z})$$

is isomorphic to

$$n \in \mathbb{Z} \mapsto 2n \in \mathbb{Z}$$

and the interval decomposition on \mathbb{R} and \mathbb{Z}_2 gives the different decomposition as follows:

$$\begin{aligned} \mathbb{X}: X_0 = \emptyset \subset X_1 = \partial M \subset X_2 = M, \\ H_1(\mathbb{X}; \mathbb{Z}_2) = I(1, 2) \oplus I(2, \infty), \quad H_1(\mathbb{X}; \mathbb{R}) = I(1, \infty). \end{aligned}$$

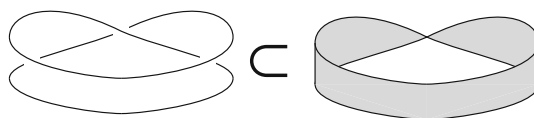


Fig. 1 Möbius strip and its boundary

In this example, both $H_1(\partial M; \mathbb{Z})$ and $H_1(M; \mathbb{Z})$ are free, but $H_1(M, \partial M; \mathbb{Z}) \simeq \mathbb{Z}_2$ and this is not free. This fact is key to the different diagrams. Section 3 shows some other examples.

We present the following theorem.

Theorem 1.6 $D_q(\mathbb{X}; \mathbb{k})$ is independent of the choice of \mathbb{k} if $H_q(X_n, X_m; \mathbb{Z})$ is free for any $0 \leq m < n \leq N$ and $H_{q-1}(X_n; \mathbb{Z})$ is free for any $0 \leq n \leq N$.

We remark that the first assumption of this theorem includes the freeness of $H_q(X_n; \mathbb{Z}) = H_q(X_n, X_0; \mathbb{Z})$ since $X_0 = \emptyset$.

This theorem yields the following corollaries.

Corollary 1.7 $D_0(\mathbb{X}; \mathbb{k})$ is always independent of the choice of \mathbb{k} .

Corollary 1.8 When \mathbb{X} is a filtration of finite cell/simplicial/cubical complexes embedded in \mathbb{R}^M , the $(M - 1)$ th persistent homology gives the same PD among any fields \mathbb{k} .

Corollary 1.7 derives from the fact that $H_{-1}(\cdot) = 0$ and $H_0(X_n, X_m; \mathbb{Z})$ is free for any $n > m$. Corollary 1.8 is proved in Sect. 4.1. The above two corollaries ensure that if a filtration is embedded in \mathbb{R}^2 , all non-trivial persistence diagrams D_0 and D_1 do not depend on the choice of the coefficient field.

We also have the following theorem which provides a sufficient condition for the freeness of $H_q(X_n, X_m; \mathbb{Z})$.

Theorem 1.9 For a given q , $H_q(X_n, X_m; \mathbb{Z})$ are free for any $0 \leq m < n \leq N$ if $D_q(\mathbb{X}; \mathbb{k})$ is independent of the choice of \mathbb{k} and $H_{q-1}(X_n; \mathbb{Z})$ is free for any $0 \leq n \leq N$.

From the above two theorems, we have the following corollary.

Corollary 1.10 Let M be a non-negative integer. $D_q(\mathbb{X}; \mathbb{k})$ is independent of the choice of \mathbb{k} for all $q = 0, \dots, M$ if and only if $H_q(X_n, X_m; \mathbb{Z})$ are free for any $0 \leq m < n \leq N$ and $q = 0, \dots, M$.

As mentioned at Theorem 1.6, the latter condition of this corollary includes the freeness of all absolute homology groups.

The above theorems and corollaries show the deep relationship between Question 1.3 and non-trivial torsion subgroups of relative homology groups. Dey et al. [11] proved the relationship between the existence of a non-trivial torsion subgroup and the computational complexity of a kind of optimization problem on homology algebra. The result showed that the problem essentially becomes harder if some relative homology groups have non-trivial torsion subgroups for the given complex. The relation between their study and our results is discussed in our conclusion section. For the proof of the above theorems and corollaries, the universal coefficient theorem for ordinary homology plays an important role. We note that the universal coefficient theorem for persistence modules has already been developed by Bubenik and Milićević [3]. Moreover, the Künneth theorem for persistence modules has been developed in the literature [3, 16, 29].

From the above discussion, the following question arises.

Question 1.11 Is there an efficient algorithm for checking the condition of Corollary 1.10?

Such an algorithm would be useful to provide information as to whether we should be concerned about field choice. To describe the algorithm, we assume the following condition.

Condition 1.12 $X = \{\sigma_1, \sigma_2, \dots, \sigma_N\}$ is a finite simplicial, cubical, or cell complex and the subset $X_k = \{\sigma_1, \dots, \sigma_k\}$ is a sub-complex of X .

With this setting, we consider the filtration of complexes $\mathbb{X}: \emptyset = X_0 \subset X_1 \subset \dots \subset X_N$. For a general finite filtration, we can construct a filtration satisfying Condition 1.12 by ordering all simplices properly.

We can compute relative homology groups for all $m < n$ on a computer, but that would be cumbersome and inefficient because the number of possible pairs (m, n) is $(N + 1)N/2$. More precisely, since the computation cost (time complexity) of determining the torsion part of $H_q(X_n, X_m; \mathbb{Z})$ is in worst case $O(n^\theta)$, $\theta \approx 2.376$,³ the total cost in the worst case under Condition 1.12 is

$$\sum_{1 \leq m < n \leq N} O(n^\theta) = O(N^{\theta+2}).$$

The following theorem is proved in Sect. 6.

Theorem 1.13 *There is an algorithm for judging the condition in Corollary 1.10 whose time complexity is the same as the algorithm for computing a PD.*

The algorithm is shown in Algorithm 6.1 in Sect. 6. Indeed, we can realize the algorithm simply by changing the coefficient field to \mathbb{Z} in the standard persistence algorithm. In Sect. 7, we apply the algorithm to some examples shown in Sect. 3 and demonstrate that it performs as expected. A performance benchmark is also covered in that section.

Question 1.14 If Algorithm 6.1 shows the dependence of the choice of the field \mathbb{k} , how can we compute persistence diagrams for multiple coefficient fields?

To answer this question, we can use the algorithms introduced in Sect. 1.1. We can also combine our new algorithm with the algorithms for multiple coefficient fields. The way to combine is explained in Sect. 6.1.

We now pose the following additional question.

Question 1.15 How often do we face filtrations with non-trivial torsion subgroups?

We can construct such an example by a Möbius strip as shown in Example 1.5, but would we often face such a filtration? To demonstrate the probability of torsions in

³ To compute the torsion subgroup of a homology group, we need to compute the Smith normal form of the boundary matrix, and the computational cost of the Smith normal form is $O(T^\theta)$ in the worst case where T is the number of simplices and $\theta \approx 2.376$ is a constant. The constant θ derives from the time complexity of the multiplication of two $T \times T$ matrices. The time complexity of persistent homology is also $O(T^\theta)$ in the worst case. See [26, 32] for further details.

filtrations, we conduct a numerical experiment for random data in \mathbb{R}^3 . From this experiment, we show that filtrations with non-trivial torsion subgroups are very rare. This suggests that, in practical terms, if the data are in \mathbb{R}^3 , we do not need to be particularly concerned about the torsion problem. We also conduct another numerical experiment for random filtrations in a high-dimensional space. This second experiment shows that the filtrations with non-trivial torsion subgroups are to be expected when the space is high dimensional. The relationship between the probability of non-trivial torsion and the dimension of the space is further investigated using Vietoris–Rips filtrations. The third experiment suggests that the larger dimension is, the larger probability is.

For Question 1.15, we remark the result by Kahle et al. [22]. The authors estimated the minimum number of vertices of a simplicial complex whose homology group has the desired torsion part. The result did not say anything about the frequency of the appearance of a non-trivial torsion subgroup, but it enables us to estimate which type of torsion subgroup is easy to realize. The following question is also important.

Question 1.16 When $D_q(\mathbb{X}; \mathbb{k})$ changes depending of the choice of \mathbb{k} , how does $D_q(\mathbb{X}; \mathbb{k})$ change?

In the above example about a Möbius strip, a long interval $I(1, \infty)$ is split into two shorter intervals, $I(1, 2)$ and $I(2, \infty)$, when \mathbb{k} changes from \mathbb{R} to \mathbb{Z}_2 . From the example, we expect that a long interval indecomposable tends to be split into shorter intervals when \mathbb{k} changes from \mathbb{R} to \mathbb{Z}_p . The following theorem proved in Sect. 9 partially answers the question. We remark that only this theorem considers a \mathbb{R} -parameterized filtration instead of an \mathbb{N} -parameterized filtration.

Theorem 1.17 *Let q be a positive integer and we consider an \mathbb{R} -parameterized filtration $\mathbb{X} = \{X_t\}_{t \in \mathbb{R}}$. Assume that X_t is finite, that is, X_t for sufficiently large t is a finite simplicial complex. We also assume that $X_t = \emptyset$ for sufficiently small t and $H_q(X_t) = 0$ for sufficiently large t . Let f be a C^2 convex function on $[0, \infty)$ with $f(0) = 0$. Then the following inequality holds if $H_q(X_t; \mathbb{Z})$ and $H_{q-1}(X_t; \mathbb{Z})$ are free for all t :*

$$\sum_{(b,d) \in D_q(\mathbb{X}; \mathbb{R})} f(d-b) \geq \sum_{(b,d) \in D_q(\mathbb{X}; \mathbb{Z}_p)} f(d-b).$$

When f is strictly convex, the equality holds if and only if $D_q(\mathbb{X}; \mathbb{R}) = D_q(\mathbb{X}; \mathbb{Z}_p)$.

If \mathbb{X} is embedded in \mathbb{R}^3 , the theorem holds for $q = 1$. Unfortunately, the freeness condition in Theorem 1.17 does not hold for higher dimensional filtrations in general. For $f(x) = x^r$ with $r > 1$, the inequality means

$$W_r(D_q(\mathbb{X}; \mathbb{R}), \emptyset) \geq W_r(D_q(\mathbb{X}; \mathbb{Z}_p), \emptyset),$$

where W_r is the r -Wasserstein distance. In some sense, the r -Wasserstein distance from the empty diagram indicates the information richness of the diagram. Therefore, $D_q(\mathbb{X}; \mathbb{R})$ contains richer information than $D_q(\mathbb{X}; \mathbb{Z}_p)$ under the condition of the theorem.

1.3 Organization of the Paper

The remainder of the paper is organized as follows. Section 2 reviews the basic concepts of persistent homology. Section 3 shows some examples which exhibit the dependence of PDs to their coefficient fields. Sections 4 and 5 prove Theorems 1.6 and 1.9. Section 6 presents an efficient algorithm which permits judgment and the proof which testifies to the correctness of the algorithm. Section 7 introduces an implementation of the algorithm in our data analysis software based on persistent homology, HomCloud.⁴ This section also shows the performance benchmark. Section 8 presents numerical experiments to measure the probability of the appearance of non-trivial torsions in random filtrations. Section 9 is the proof of Theorem 1.17 and, finally, conclusions are offered in Sect. 10.

2 Persistent Homology

In this section, we present some fundamental concepts for persistent homology.

2.1 Filtrations

A filtration is an increasing sequence of topological spaces. One typical filtration is the union of r -balls constructed from a pointcloud in \mathbb{R}^M . For a pointcloud, a set of finite points $\{x_i\}$, X_r is defined as

$$X_r = \bigcup_i B_{x_i}(r),$$

where $B_x(r)$ is the closed ball whose center is x and radius is r .⁵ The sequence of X_r parameterized by r , $\{X_r\}_{r \geq 0}$, is obviously a filtration. This filtration is used to investigate the shape formed by the pointcloud.

For a practical application of persistent homology, we usually use finite simplicial or cubical filtrations since such filtrations are practical to consider on a computer. One well-known filtration is the Čech filtration. The *Čech complex* $\check{C}ech(P, r)$ of a pointcloud $P = \{x_i\}$ with radius parameter $r \geq 0$ is defined as follows:

$$\check{C}ech(P, r) = \left\{ \{x_{i_1}, \dots, x_{i_k}\} \subset P \mid \bigcap_{n=1}^k B_{x_{i_n}}(r) \neq \emptyset \right\}.$$

The filtration $\{\check{C}ech(P, r)\}_{r \geq 0}$ is called a *Čech filtration*. From the nerve theorem, $\check{C}ech(P, r)$ is homotopy equivalent to $\bigcup_i B_{x_i}(r)$ and we can use the Čech filtration to investigate the union of r -balls. There are many simplices in a Čech complex

⁴ <https://homcloud.dev>.

⁵ Open balls are usually used to define Čech filtrations. The nerve theorem holds for both open and closed balls.

for a large pointcloud and we usually use an *alpha complex* [12, 15] instead, since the alpha complex is homotopy equivalent to the Čech complex and the number of simplices of the alpha complex is much smaller than the Čech complex. The alpha complex has another advantage in that it can be embedded in \mathbb{R}^M but such embedding is usually impossible for the Čech complex. Since the number of points is finite, the Čech filtration and alpha filtration is bounded. Therefore we can regard these \mathbb{R} -parameterized filtrations as \mathbb{N} -parameterized filtrations.

2.2 Computation of a Persistence Diagram

Under Condition 1.12, Algorithm 2.1 computes the PD of the filtration [14, 27, 33]. To simplify the algorithm, all simplices of all dimensions are mixed, and in the output, all birth-death pairs of all degrees are also mixed. In this algorithm, $L_B(j)$ means

$$L_B(j) = \begin{cases} \max \{i \mid B_{ij} \neq 0\} & \text{if column } j \text{ of } B \text{ is nonzero,} \\ -\infty & \text{if column } j \text{ of } B \text{ is zero,} \end{cases} \tag{1}$$

where B is a matrix and j is an integer. Furthermore, in this algorithm, matrix B is reduced from left column to right column. After terminating the algorithm, the PD is computed as follows:

$$D(\mathbb{X}) = \{(L_{\hat{B}}(j), j) \mid L_{\hat{B}}(j) \neq -\infty\} \cup \{(j, \infty) \mid L_{\hat{B}}(j) = -\infty \text{ and } \forall i, L_{\hat{B}}(i) \neq j\},$$

where \hat{B} is the matrix returned by the algorithm. The q th PD is given from $D(\mathbb{X})$ as follows:

$$D_q(\mathbb{X}) = \{(i, j) \in D(\mathbb{X}) \mid \dim \sigma_i = q\}.$$

Algorithm 2.1 Algorithm to compute persistence diagrams

```

B ← the boundary matrix with respect to the basis  $\{\sigma_1, \dots, \sigma_N\}$ 
for  $j = 1, \dots, N$  do
    while there exists  $i < j$  with  $L_B(i) = L_B(j) \neq -\infty$  do
        let  $s = B_{L_B(j),j} / B_{L_B(i),i}$ 
        add  $(-s) \times (\text{column } i)$  to column  $j$  of  $B$  ▷ left-to-right reduction
return  $B$ 

```

Indeed, Theorem 1.13 shows that the algorithm for judging the condition of Corollary 1.10 is given by restricting Algorithm 2.1 to integer coefficients. Therefore, the time complexity of the Theorem 1.13 algorithm is as per Algorithm 2.1.

2.3 Persistent Betti Number

From the definition of a PD, we have the following relationship between the map $H_q(X_m; \mathbb{k}) \rightarrow H_q(X_n; \mathbb{k})$ and a PD:

$$\begin{aligned} \beta_m^n(\mathbb{k}) &:= \text{rank}(H_q(X_m; \mathbb{k}) \rightarrow H_q(X_n; \mathbb{k})) \\ &= \#\{(b, d) \in D_q(\mathbb{X}; \mathbb{k}) \mid b \leq m \leq n < d\}. \end{aligned}$$

This $\beta_m^n(\mathbb{k})$ is called a *persistent Betti number* or a *rank invariant*. Hence, the following identity holds:

$$(\text{multiplicity of } D_q(\mathbb{X}; \mathbb{k}) \text{ at } (b, d)) = \beta_b^{d-1}(\mathbb{k}) - \beta_{b-1}^{d-1}(\mathbb{k}) - \beta_b^d(\mathbb{k}) + \beta_{b-1}^d(\mathbb{k}).$$

When $d = \infty$, the following equation holds instead:

$$(\text{multiplicity of } D_q(\mathbb{X}; \mathbb{k}) \text{ at } (b, \infty)) = \beta_b^N(\mathbb{k}) - \beta_{b-1}^N(\mathbb{k}).$$

The next lemma follows directly from the foregoing.

Lemma 2.2 $D_q(\mathbb{X}; \mathbb{k}) = D_q(\mathbb{X}; \mathbb{k}')$ if and only if $\beta_m^n(\mathbb{k}) = \beta_m^n(\mathbb{k}')$ for all $0 \leq m \leq n \leq N$.

2.4 Universal Coefficient Theorem

The universal coefficient theorem is fundamental for homology theory and plays an important role in this paper. We review the theorem here to prepare for what follows.

Theorem 2.3 (universal coefficient theorem [17]) *Let X be a topological space, \mathbb{k} a ring, and $q \geq 0$. The following sequence is a natural short exact sequence:*

$$0 \rightarrow H_q(X; \mathbb{Z}) \otimes \mathbb{k} \rightarrow H_q(X; \mathbb{k}) \rightarrow \text{Tor}(H_{q-1}(X; \mathbb{Z}), \mathbb{k}) \rightarrow 0.$$

Furthermore, this sequence splits, though not naturally.

We use the above theorem in the following form.

Theorem 2.4 *Let X and Y be topological spaces, $f: X \rightarrow Y$ a continuous map, \mathbb{k} a ring, and $q \geq 0$. If $H_{q-1}(X; \mathbb{Z})$ and $H_{q-1}(Y; \mathbb{Z})$ are free, the following commutative diagram holds:*

$$\begin{array}{ccc} H_q(X; \mathbb{Z}) \otimes \mathbb{k} & \xrightarrow{\cong} & H_q(X; \mathbb{k}) \\ f_* \otimes \text{id}_{\mathbb{k}} \downarrow & \circlearrowleft & f_* \downarrow \\ H_q(Y; \mathbb{Z}) \otimes \mathbb{k} & \xrightarrow{\cong} & H_q(Y; \mathbb{k}). \end{array}$$

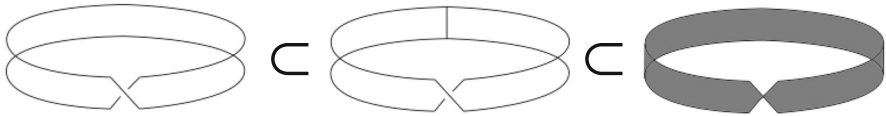


Fig. 2 Visualization of \mathbb{X}

This theorem states that the induced map $f_* : H_q(X; \mathbb{k}) \rightarrow H_q(Y; \mathbb{k})$ is completely described by $f_* \otimes \text{id}_{\mathbb{k}} : H_q(X; \mathbb{Z}) \otimes \mathbb{k} \rightarrow H_q(Y; \mathbb{Z}) \otimes \mathbb{k}$ if $H_{q-1}(X; \mathbb{Z})$ and $H_{q-1}(Y; \mathbb{Z})$ are free. We use the theorem for an inclusion map between simplicial/cell/cubical complexes.

3 Examples of Diagrammatic Changes Induced by Coefficient Field Changes

In this section, we give some examples of persistent homology, whose interval decomposition depends on the choice of coefficient field.

Example 3.1 Let S^1 be a circle and $S^1 \vee S^1$ a bouquet of 2-circles. We consider a filtration $\mathbb{X} : \emptyset \rightarrow S^1 \xrightarrow{f} S^1 \vee S^1 \xrightarrow{g} Y$ as in Fig. 2, where Y is a space homotopy equivalent to S^1 ,

$$H_1(f) = \begin{pmatrix} 1 \\ 1 \end{pmatrix}, \quad H_1(g) = (1 \ 1).$$

By taking the 1st homology of this filtration, we obtain the 1st persistent homology

$$H_1(\mathbb{X}; \mathbb{Z}) = 0 \longrightarrow \mathbb{Z} \xrightarrow{\begin{pmatrix} 1 \\ 1 \end{pmatrix}} \mathbb{Z}^2 \xrightarrow{(1 \ 1)} \mathbb{Z},$$

with a coefficient ring \mathbb{Z} . Then $H_1(\mathbb{X}; \mathbb{Z}_2) = H_1(\mathbb{X}; \mathbb{Z}) \otimes_{\mathbb{Z}} \mathbb{Z}_2$ has the interval decomposition $I(1, 2) \oplus I(2, 3)$. On the other hand, $H_1(\mathbb{X}; \mathbb{R}) = H_1(\mathbb{X}; \mathbb{Z}) \otimes_{\mathbb{Z}} \mathbb{R}$ has the interval decomposition $I(1, 3) \oplus I(2, 2)$. Thus, the interval decomposition of the 1st persistent homology of \mathbb{X} depends on the choice of coefficient field.

Note that if we consider a bouquet of p -circles for a prime p , then we obtain the 1st persistent homology, which has different decompositions over \mathbb{Z}_p and \mathbb{R} .

By using Example 3.1, we can consider the 1st persistent homology, whose interval decomposition depends on the choice of characteristic $p > 0$.

Example 3.2 We can construct a more complicated example in a similar way. Let

$$M = 0 \longrightarrow \mathbb{Z} \xrightarrow{\begin{pmatrix} 1 \\ 1 \end{pmatrix}} \mathbb{Z}^2 \xrightarrow{(1 \ 1)} \mathbb{Z} \xrightarrow{\begin{pmatrix} 1 \\ 1 \\ 1 \end{pmatrix}} \mathbb{Z}^3 \xrightarrow{(1 \ 1 \ 1)} \mathbb{Z}$$

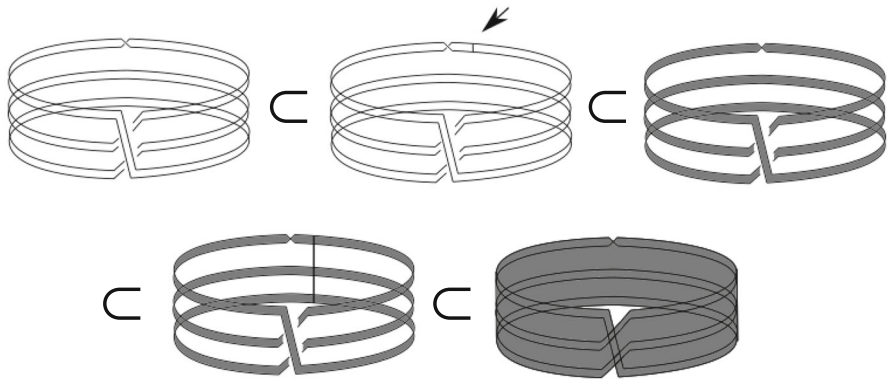


Fig. 3 Visualization of \mathbb{X} with $M = H_1(\mathbb{X})$

be the 1st \mathbb{Z} -persistent homology coming from the filtration of Fig. 3. Then M has the following interval decomposition:

$$M \cong \begin{cases} I(1, 2) \oplus I(2, 5) \oplus I(4, 4)^2 & \text{if } p = 2, \\ I(1, 4) \oplus I(2, 2) \oplus I(4, 4) \oplus I(4, 5) & \text{if } p = 3. \end{cases}$$

Example 3.3 Other examples are double and triple loop pointclouds. Figure 4(a) shows the double loop pointcloud. The pointcloud is located on the boundary of a Möbius strip. We compute the 1st PDs of the double loop pointcloud with fields $\mathbb{Z}_2, \mathbb{Z}_3,$ and \mathbb{Z}_5 . The alpha filtration of the pointcloud is used for the computation. The diagrams are shown in Fig. 4, (b)–(d). Note that (c) and (d) are the same diagram. The difference between (b) and (c) is only two birth-death pairs. Figure 4(e) shows the triple loop pointcloud and (f)–(h) show the 1st PDs of the triple loop pointcloud with fields $\mathbb{Z}_2, \mathbb{Z}_3,$ and \mathbb{Z}_5 . To be expected, (f) and (h) are the same diagram and (g) is different from (f) and (h).

4 Proof of Theorem 1.6

The following lemma is required to prove the theorem.

Lemma 4.1 *If $H_q(X_n, X_m; \mathbb{Z})$ is free, $\text{coker}(\phi_m^n : H_q(X_m; \mathbb{Z}) \rightarrow H_q(X_n; \mathbb{Z}))$ is also free.*

Proof We have the following long exact sequence [17, Theorem 2.16, p. 117] for the pair (X_n, X_m) :

$$\dots \rightarrow H_q(X_m; \mathbb{Z}) \xrightarrow{\phi_m^n} H_q(X_n; \mathbb{Z}) \xrightarrow{\psi_m^n} H_q(X_n, X_m; \mathbb{Z}) \rightarrow \dots,$$

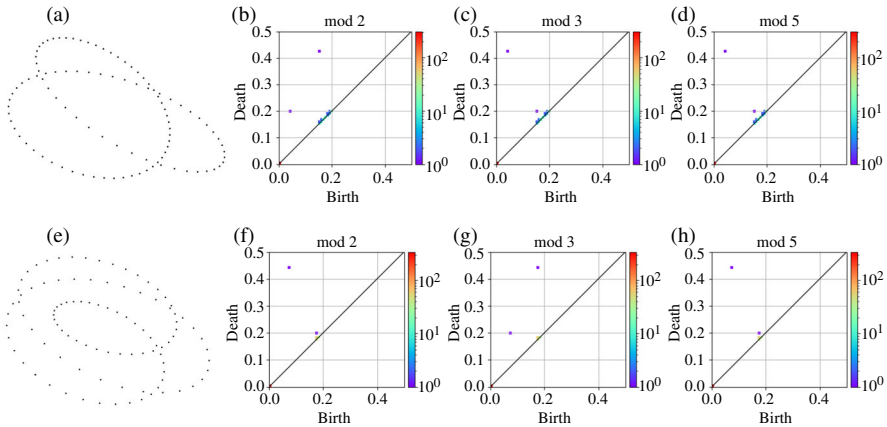


Fig. 4 1st PDs with various fields for double and triple loop pointclouds. (a) A double loop point cloud. (b) The PD of the double loop with \mathbb{Z}_2 . (c) The PD of the double loop with \mathbb{Z}_3 . (d) The PD of the double loop with \mathbb{Z}_5 . (e) A triple loop point cloud. (f) The PD of the triple loop with \mathbb{Z}_2 . (g) The PD of the triple loop with \mathbb{Z}_3 . (h) The PD of the triple loop with \mathbb{Z}_5

where ψ_m^n is induced by canonical projection. Therefore, we have the following relationship between coker ϕ_m^n and $H_q(X_n, X_m; \mathbb{Z})$:

$$\text{coker } \phi_m^n = H_q(X_n; \mathbb{Z}) / \text{im } \phi_m^n = H_q(X_n; \mathbb{Z}) / \ker \psi_m^n \simeq \text{im } \psi_m^n \subset H_q(X_n, X_m; \mathbb{Z}).$$

To complete the proof, we need to show that $\text{im } \psi_m^n$ is free, and this derives from the following well-known theorem.

Theorem 4.2 *Any sub-module of a free \mathbb{Z} -module is also free.*

By assumption, $H_q(X_n, X_m; \mathbb{Z})$ is free, and then $\text{coker } \phi_m^n \simeq \text{im } \psi_m^n$ is free by Theorem 4.2. □

We also need a discussion on Smith normal form for the proof of the theorem. When $H_q(X_m; \mathbb{Z})$ and $H_q(X_n; \mathbb{Z})$ are free, ϕ_m^n is a homomorphism between two finitely generated free \mathbb{Z} -modules and the map has a Smith normal form (SNF). That is, by taking an appropriate basis, ϕ_m^n can be represented by the following \mathbb{Z} matrix:

$$\begin{pmatrix} \alpha_1 & 0 & 0 & \cdots & 0 \\ 0 & \alpha_2 & 0 & \cdots & 0 \\ & & 0 & \ddots & \\ \vdots & \vdots & & \ddots & \\ 0 & 0 & & & \alpha_K \\ & & & & & O \\ & & & & & & O \end{pmatrix}, \tag{2}$$

where $0 < \alpha_k \in \mathbb{Z}$ and $\alpha_k \mid \alpha_{k+1}$ for any k . We will show the following two lemmas about SNF.

Lemma 4.3 *Assume that $H_q(X_n; \mathbb{Z})$, $H_q(X_m; \mathbb{Z})$, $H_{q-1}(X_n; \mathbb{Z})$, and $H_{q-1}(X_m; \mathbb{Z})$ are free. Then $\beta_m^n(\mathbb{k})$ is independent of the choice of \mathbb{k} if and only if $\alpha_1 = \dots = \alpha_K = 1$ in (2).*

Proof From Theorem 2.4, the following relationship holds:

$$\begin{aligned} \beta_m^n(\mathbb{k}) &= \text{rank}(\phi_m^n : H_q(X_m; \mathbb{k}) \rightarrow H_q(X_n; \mathbb{k})) \\ &= \text{rank}(\phi_m^n \otimes \text{id}_{\mathbb{k}} : H_q(X_m; \mathbb{Z}) \otimes \mathbb{k} \rightarrow H_q(X_n; \mathbb{Z}) \otimes \mathbb{k}). \end{aligned}$$

From the above equation and (2), we know that $\beta_m^n(\mathbb{k})$ is independent of the choice of \mathbb{k} if and only if $\alpha_1 = \dots = \alpha_K = 1$. □

Lemma 4.4 *Assume that $H_q(X_n; \mathbb{Z})$ and $H_q(X_m; \mathbb{Z})$ are free. Then $\text{coker } \phi_m^n$ is free if and only if $\alpha_1 = \dots = \alpha_K = 1$ in (2).*

Proof From SNF, we also have the following:

$$\text{coker } \phi_m^n \simeq \bigoplus_{k=k_0}^K (\mathbb{Z}/\alpha_k \mathbb{Z}) \oplus \mathbb{Z}^{L-K},$$

where $k_0 = 1 + \max \{i \mid \alpha_i = 1\}$ and $L = \text{rank } H_q(X_n; \mathbb{Z})$. We immediately lead to the conclusion of Lemma 4.4. □

Using Lemmas 2.2, 4.1, 4.3, and 4.4, we can immediately show the independence of $D_q(\mathbb{X}; \mathbb{k})$ from the choice of \mathbb{k} under the following assumptions:

- $H_q(X_n, X_m; \mathbb{Z})$ is free for any $0 \leq m < n \leq N$. This condition includes that $H_q(X_n; \mathbb{Z})$ is free for any n .
- $H_{q-1}(X_n; \mathbb{Z})$ is free for any $0 \leq n \leq N$.

4.1 Proof of Corollary 1.8

Using Theorem 1.6, we prove the independence of $D_{M-1}(\mathbb{X}; \mathbb{k})$ from the choice of \mathbb{k} when the filtration \mathbb{X} is embedded in \mathbb{R}^M .

Proof of Corollary 1.8 First, from standard homology theory [17, Corollary 3.46, p.256], $H_{M-1}(X_n; \mathbb{Z})$ and $H_{M-2}(X_n; \mathbb{Z})$ are free since $X_n \subset \mathbb{R}^M$. Second, the Alexander duality for relative homology [31, Theorem 6.2.15] gives the following isomorphism:

$$H_{M-1}(X_n, X_m; \mathbb{Z}) \simeq H^1(\bar{X}_m, \bar{X}_n; \mathbb{Z}),$$

where $\bar{X}_m = \mathbb{R}^M \setminus X_m$ and $\bar{X}_n = \mathbb{R}^M \setminus X_n$. In addition, from universal coefficient theorem for relative cohomology [17, Theorem 3.2] gives the following isomorphism:

$$H^1(\bar{X}_m, \bar{X}_n; \mathbb{Z}) \simeq \text{Ext}(H_0(\bar{X}_m, \bar{X}_n; \mathbb{Z}), \mathbb{Z}) \oplus \text{Hom}_{\mathbb{Z}}(H_1(\bar{X}_m, \bar{X}_n; \mathbb{Z}), \mathbb{Z}). \quad (3)$$

Since $H_0(\bar{X}_m, \bar{X}_n; \mathbb{Z})$ is free, $\text{Ext}(H_0(\bar{X}_m, \bar{X}_n; \mathbb{Z}), \mathbb{Z}) = 0$. In addition,

$$\text{Hom}_{\mathbb{Z}}(H_1(\bar{X}_m, \bar{X}_m; \mathbb{Z}), \mathbb{Z})$$

is isomorphic to the free part of $H_1(\bar{X}_m, \bar{X}_m; \mathbb{Z})$. Therefore, $H_{M-1}(X_n, X_m; \mathbb{Z}) \simeq H^1(\bar{X}_m, \bar{X}_n; \mathbb{Z})$ is free by the isomorphism (3). Thus, the assumption of Theorem 1.6 is satisfied, and hence $D_{M-1}(\mathbb{X}; \mathbb{k})$ is independent of the choice of \mathbb{k} . \square

5 Proof of Theorem 1.9 and Corollary 1.10

The proof is similar to that of Theorem 1.6, but slightly more complex. We need the following proposition.

Proposition 5.1 *$H_q(X_n, X_m; \mathbb{Z})$ is free if $\text{coker}(\phi_m^n : H_q(X_m; \mathbb{Z}) \rightarrow H_q(X_n; \mathbb{Z}))$ and $H_{q-1}(X_m; \mathbb{Z})$ are free.*

Proof From the long exact sequence for the pair (X_n, X_m) ,

$$\dots \rightarrow H_q(X_m; \mathbb{Z}) \xrightarrow{\phi_m^n} H_q(X_n; \mathbb{Z}) \xrightarrow{\psi_m^n} H_q(X_n, X_m; \mathbb{Z}) \xrightarrow{\partial} H_{q-1}(X_m; \mathbb{Z}) \rightarrow \dots,$$

we have the following facts:

$$\begin{aligned} \text{coker } \phi_m^n &\simeq \text{im } \psi_m^n \subset H_q(X_n, X_m; \mathbb{Z}), \\ \text{im } \partial &\simeq H_q(X_n, X_m; \mathbb{Z}) / \text{im } \psi_m^n. \end{aligned}$$

$\text{im } \partial$ is free since $H_{q-1}(X_m)$ is free. We complete the proof by the following theorem from standard algebra.

Theorem 5.2 *Let M be a module over \mathbb{Z} and N be a sub-module of M . M is finitely generated and free if N and M/N are finitely generated and free.*

Proof We have an exact sequence

$$0 \rightarrow N \rightarrow M \rightarrow M/N \rightarrow 0.$$

Since M/N is free (in particular, it is projective), the exact sequence is split. That is, $M \cong N \oplus M/N$, and hence M is finitely generated and free. \square

Proposition 5.1 has been proved. \square

Proof of Theorem 1.9 We assume that $D_q(\mathbb{X}; \mathbb{k})$ is independent of the choice of \mathbb{k} and $H_{q-1}(X_n; \mathbb{Z})$ is free for any $0 \leq n \leq N$. Then from Lemma 2.2, $\beta_m^n(\mathbb{k})$ is independent of \mathbb{k} for any m and n . Especially for any n and q , $\beta_n^n(\mathbb{k}) = \dim H_q(X_n; \mathbb{k})$ is independent of \mathbb{k} and therefore $H_q(X_n; \mathbb{Z})$ is free due to the universal coefficient theorem since $H_{q-1}(X_n; \mathbb{Z})$ is free. Then $\text{coker } \phi_m^n$ is free because of Lemma 4.4. From the above fact and Proposition 5.1, we conclude that $H_q(X_n, X_m; \mathbb{Z})$ is free. \square

Proof of Corollary 1.10 Here, we will show that the following two conditions are equivalent.

- (a) $D_q(\mathbb{X}; \mathbb{k})$ is independent of the choice of \mathbb{k} for all $q = 0, \dots, M$.
- (b) $H_q(X_n, X_m; \mathbb{Z})$ are free for any $0 \leq m < n \leq N$, and $q = 0, \dots, M$.

From Theorem 1.6, it is straightforward to show that (b) implies (a). We can show the converse by induction on q . For $q = 0$, it is trivial that $H_0(X_n, X_m; \mathbb{Z})$ is free and the induction process proceeds by using Theorem 1.9. □

6 Algorithm to Determine the Dependence of $D_q(\mathbb{X}; \mathbb{k})$ on \mathbb{k}

In this section, we explore an algorithm to judge the dependence of $D_q(\mathbb{X}; \mathbb{k})$ on \mathbb{k} . The exact description of Theorem 1.13 is as follows.

- If Algorithm 6.1 returns “independent”, $D_q(\mathbb{X}; \mathbb{k})$ is independent of the choice of \mathbb{k} .
- If Algorithm 6.1 returns “dependent”, $D_q(\mathbb{X}; \mathbb{k})$ depends on the choice of \mathbb{k} .

Algorithm 6.1 Algorithm to determine the dependence of $D_q(\mathbb{X}; \mathbb{k})$ on \mathbb{k}

```

let  $B$  be the matrix representation of the boundary operator
for  $j = 1, \dots, N$  do ▷ (OUTERLOOP)
  while there exists  $i < j$  with  $L_B(i) = L_B(j) \neq -\infty$  do ▷ (INNERLOOP)
    let  $s = -B_{L_B(j),j} / B_{L_B(i),i}$  ▷ (A)
    add  $s \times$  (column  $i$ ) to column  $j$  of  $B$  ▷ left-to-right reduction
  if  $L(j) \neq -\infty$  and  $B_{L_B(j),j} \notin \{\pm 1\}$  then ▷ (B)
    print  $|B_{L_B(j),j}|$ 
    return “dependent”
return “independent”
    
```

In Algorithm 6.1, L_B means (1). We remark that $B_{L_B(i),i}$ in this algorithm is always ± 1 at (A), so the division at (A) always applies. This is because the condition is checked at (B).

We use the following notation.

Notation 6.2 – R is \mathbb{Z} or a field.

- $C(X_k; R) := \bigoplus_{q=0}^{\dim \mathbb{X}} C_q(X_k; R)$.
- $\partial_k: C(X_k; R) \rightarrow C(X_k; R) = \bigoplus_{q=0}^{\dim \mathbb{X}} (\partial_k^{(q)}: C_q(X_k; R) \rightarrow C_{q-1}(X_k; R))$.
- D_R is the boundary matrix of ∂_N with respect to the basis $\{\sigma_1, \dots, \sigma_N\}$.

- $F_{\mathbb{k}}(AB) = F_{\mathbb{k}}(A)F_{\mathbb{k}}(B)$ for any \mathbb{Z} -matrices A, B .
- $F_{\mathbb{k}}(D_{\mathbb{Z}}) = D_{\mathbb{k}}$.
- $F_{\mathbb{k}}(U_{ij}(s)) = U_{ij}(F_{\mathbb{k}}(s))$. That is, $F_{\mathbb{k}}$ maps a left-to-right reduction \mathbb{Z} -matrix to a left-to-right reduction \mathbb{k} -matrix.
- If B is an n -reduced \mathbb{Z} -matrix and $B_{L_B(i),i} \in \{\pm 1\}$ for all i with $L_B(i) \neq -\infty$, $F_{\mathbb{k}}(B)$ is an n -reduced \mathbb{k} -matrix.

First, we prove that $D_{\mathbb{k}}(X; \mathbb{k})$ is independent of the choice of \mathbb{k} when the algorithm returns “independent”. Let \hat{B} be matrix B in Algorithm 6.1 when the program terminates. Since the left-to-right reduction in Algorithm 6.1 is equivalent to the multiplication of $U_{ij}(s)$ from right, \hat{B} can be written as

$$\hat{B} = D_{\mathbb{Z}}U_1 \cdots U_r, \tag{4}$$

where U_1, \dots, U_r is left-to-right reduction \mathbb{Z} -matrices. Therefore we have

$$F_{\mathbb{k}}(\hat{B}) = D_{\mathbb{k}}F_{\mathbb{k}}(U_1) \cdots F_{\mathbb{k}}(U_r). \tag{5}$$

From the terminating condition of the while loop (INNERLOOP) in Algorithm 6.1, \hat{B} is N -reduced. Moreover, since Algorithm 6.1 checks condition (B), $\hat{B}_{L_{\hat{B}}(j),j} \in \{\pm 1\}$ for any j with $L_{\hat{B}}(j) \neq -\infty$ and $F_{\mathbb{k}}(\hat{B})$ is N -reduced. Therefore, from pairing lemma, we conclude that the persistence diagram does not depend on the choice of \mathbb{k} .

Next we prove $D_{\mathbb{k}}(X; \mathbb{k})$ depends on the choice of \mathbb{k} if the algorithm returns “dependent”. In that case, condition (B) in the algorithm is true for one j , so let n be that j and \hat{B} be B at that time. For the same reason as the independent case, \hat{B} can be written as (4) and therefore (5) holds also in this case. At the same time,

$$L_{F_{\mathbb{R}}(\hat{B})}(n) = L_{\hat{B}}(n) > L_{F_{\mathbb{Z}_p}(\hat{B})}(n),$$

where p is a prime divisor of $|\hat{B}_{L_{\hat{B}}(n),n}|$ since $F_{\mathbb{Z}_p}(\hat{B}_{L_{\hat{B}}(n),n}) = 0$ but $F_{\mathbb{R}}(\hat{B}_{L_{\hat{B}}(n),n}) \neq 0$. Therefore $r(m, n; F_{\mathbb{R}}(\hat{B})) = 1$ but $r(m, n; F_{\mathbb{Z}_p}(\hat{B})) = 0$ where $m = L_{\hat{B}}(n)$, since $F_{\mathbb{R}}(\hat{B})$ is n -reduced, $F_{\mathbb{Z}_p}(\hat{B})$ is $(n - 1)$ -reduced, $m > L_{F_{\mathbb{Z}_p}(\hat{B})}(n)$, and $L_{F_{\mathbb{Z}_p}(\hat{B})}(j) \neq m$ for any $j < n$. From the pairing lemma, we conclude that $D(\mathbb{k}; \mathbb{Z}_p) \neq D(\mathbb{k}; \mathbb{R})$. □

6.1 Combining Algorithm 6.1 with the Modular Reconstruction Algorithm or Omni-Field Persistence

We can combine our algorithm and the modular reconstruction algorithm in the following way. Before stopping Algorithm 6.1, the two algorithms do the same matrix reductions operations, and the following equality always holds until stopping:

$$B_{6.1} \bmod (q_1 \cdots q_r) = B_{\text{mra}}.$$

Table 1 Performance benchmark results

	5000 points	50,000 points
Phat	0.0282 s	0.446 s
HomCloud	0.0323 s	0.550 s

where $B_{6,1}$ is the matrix in Algorithm 6.1 and B_{mra} is the matrix in modular reconstruction algorithm. Therefore Algorithm 6.1 can enter the modular reconstruction algorithm by applying $\text{mod}(q_1 \cdots q_r)$ to the \mathbb{Z} -matrix just after stopping Algorithm 6.1. The idea enables us to share the result of the computation with the modular reconstruction algorithm to reduce the required computational resources. We can also combine our algorithm and omni-field persistence algorithm in a similar way to the modular reconstruction algorithm using embedding homeomorphism $\mathbb{Z} \rightarrow \mathbb{Q}$.

7 Algorithm Implementation

The judgment algorithm is implemented in our software HomCloud. The twist algorithm introduced by Chen and Kerber [8] is used for faster computations. Since most elements of each column are usually zero, only non-zero values and corresponding indices are stored in two arrays. The indices are stored in ascending order and reduce the columns like merge process in merge sort [24]. The program correctly judges the existence of the torsion in the filtration given by the pointclouds shown in Fig. 4, (a) and (e).

7.1 Performance Benchmark

In this section, we explore the performance of Algorithm 6.1. We compare the program implemented in HomCloud and Phat [1].⁷ The Phat code is straightforward and efficient. The input filtration for the performance comparison is an alpha filtration constructed from random 5000 and 50,000 points in \mathbb{R}^3 . Five trials were undertaken and the average computation time is shown. In Phat, we use twist-algorithm with `bit_tree_pivot_column`, as recommended by Bauer et al. [1]. The benchmark is executed on a PC with a 1.5 GHz Intel(R) Core(TM) i7-8500Y CPU, 16 GB of memory, and the Debian 10.0 operating system. Both programs run on a single core. Results are shown in Table 1.

According to the benchmark, our new program is ca. $\times 1.20$ slower than Phat. Phat uses \mathbb{Z}_2 as a coefficient field and implements fast arithmetic operations by using bit-wise operations. The technique likely renders Phat faster and we conclude that the performance of our program is roughly as efficient as Phat.

⁷ <https://bitbucket.org/phat-code/phat/>.

8 Probability of Torsion Appearance in Random Filtrations

Here we measured the probability of the appearance of torsions of random filtrations by a numerical experiment. Corollaries 1.7 and 1.8 already ensure the independence of persistence diagrams from \mathbb{k} for a filtration embedded in \mathbb{R}^2 . Therefore, we started from filtrations in \mathbb{R}^3 .

We generated a random filtration from a pointcloud sampled from a Poisson point process in $[0, 1]^3$.⁸ The average number of points is 1000. Thus, a random number k is sampled from the Poisson distribution whose parameter is 1000, and k points are uniformly randomly sampled in $[0, 1]^3$. An alpha filtration was computed from the generated pointcloud and the condition was judged by HomCloud. Here, 10,000 trials were carried out. Only one filtration had non-trivial torsion; thus, 9999 filtrations had trivial torsion.⁹ In sum, it can be stated that a filtration with non-trivial torsion is possible but very rare. This numerical experiment suggests that there is some mathematical mechanism explaining why a random filtration with non-trivial torsion is quite rare. Exploring this further here is beyond the scope of the current paper.

In contrast, Kahle et al. [22] experimentally showed that torsion subgroups often appeared in random d -complex $Y \sim Y_d(n, p)$, introduced by Linial and Meshulam [25]. We apply our algorithm to a Linial–Meshulam process, a natural extension of Linial and Meshulam’s random complex, introduced by Hiraoka and Shirai [19]. Let $\bar{Y}(n)$ be a simplex on n vertices and let Y_0 be the $(d - 1)$ -skeleton of $\bar{Y}(n)$. Y_k for $k = 1, \dots, m$ is randomly generated by adding a d -simplex to Y_{k-1} . The d -simplex is uniformly randomly sampled from all d -simplices in $\bar{Y}(n) \setminus Y_{k-1}$. We apply the algorithm to the filtration $Y_0 \subset Y_1 \subset \dots \subset Y_m$. We used $d = 2, n = 75, m = 5000$. The number of random filtrations was 10,000. In the experiment, we found that all 10,000 random filtrations have non-trivial torsion.

The above two experiments are contrasting. We expect that the difference emanates from the dimension of the space. In the first experiment, a filtration is in \mathbb{R}^3 and in the second experiment, $\bar{Y}(75)$ has a high dimensional geometric structure. The experiments suggest that a high-dimensional random filtration has more non-trivial torsion subgroups in the relative homology groups than a filtration with a low dimensional geometric structure.

To further investigate the relationship between the dimension of the space and the non-trivial torsion in filtrations, we numerically experimented on random Vietoris–Rips filtrations. Vietoris–Rips filtrations were used since it is difficult to construct an alpha filtration of a pointcloud in a high-dimensional space. In one trial of the experiment, 1000 points are uniformly randomly sampled in \mathbb{R}^n and we judged the existence or non-existence of non-trivial torsion subgroup in the 1st persistent homology of the Vietoris–Rips filtration of the pointcloud using our algorithm. To reduce

⁸ We also try uniform random sampling of a fixed number of points (1000 points) and the result is consistent with this experiment.

⁹ Run-time errors occurred two times in these 10,000 trials. When an error occurred, we disposed of the input data and retried random sampling. The cause of the errors is probably the violation of the general position condition of the randomly generated pointcloud.

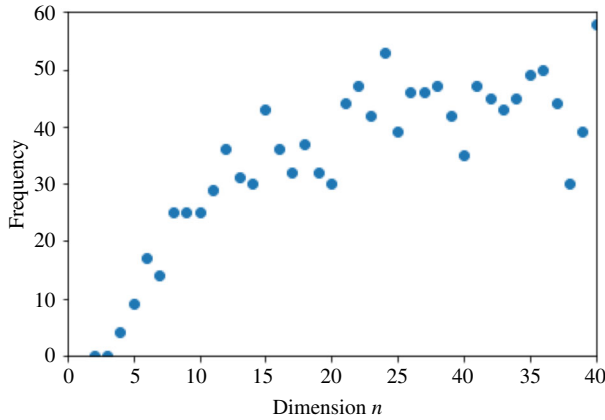


Fig. 5 Scatter plot of dimension n versus frequency in 1000 trials

the cost of computation we take the threshold of maximum edge length. The threshold is statistically determined to averagely include 166,500 edges (1/6 edges of all edges of 999-simplex) in the filtration. We determined the threshold rule by considering the limitation of our computer resources. We performed 1000 trials for each n and counted frequencies for the non-trivial torsion subgroup.

Figure 5 shows the frequencies for $n = 3, 4, \dots, 40$. We also experimented with larger thresholds to examine the validity of the threshold rule and the results were consistent with the 1/6 rule experiments. The results show that the frequency appears to monotonically increase with n and the speed of increase becomes slower as n increases.

Overall, the experiment also suggests that we do not need to be concerned about the coefficient field in most cases if the space is \mathbb{R}^3 . Thus, if there are concerns about the field choice problem in future research contexts, our proposed algorithm would be helpful.

9 Proof of Theorem 1.17

In this section, we show the following inequality for a convex C^2 function f with $f(0) = 0$ under the assumption that $H_q(X_t; \mathbb{Z})$ and $H_{q-1}(X_t; \mathbb{Z})$ are free for all t and $H_q(X_t; \mathbb{Z}) = 0$ for sufficiently large and small t :

$$\sum_{(b,d) \in D_q(\mathbb{X}; \mathbb{R})} f(d - b) \geq \sum_{(b,d) \in D_q(\mathbb{X}; \mathbb{Z}_p)} f(d - b).$$

In addition, we assume the following condition.

Condition 9.1 There exist $0 = r_0 < r_1 < \dots < r_N < r_{N+1} = \infty$ such that $r_k \leq r < r_{k+1}$ implies $X_r = X_{r_k}$.

This means that the filtration is assumed to be right-continuous. Condition 9.1 is not essential and we can prove the theorem if the filtration is left-continuous. We assume right-continuous for expository purposes.

Proof of Theorem 1.17 From Condition 9.1 and $H_q(X_t) = 0$ for sufficiently small and sufficiently large t , all birth-death pairs can be written as (r_k, r_ℓ) for $0 \leq k < \ell \leq N$. Hence, using persistent Betti numbers, we have the following equation:

$$\begin{aligned} & \sum_{(b,d) \in D_q(\mathbb{X}; \mathbb{R})} f(d - b) \\ &= \sum_{0 \leq k < \ell \leq N} f(r_\ell - r_k)(\beta_{r_k}^{r_{\ell-1}}(\mathbb{k}) - \beta_{r_{k-1}}^{r_{\ell-1}}(\mathbb{k}) - \beta_{r_k}^{r_\ell}(\mathbb{k}) + \beta_{r_{k-1}}^{r_\ell}(\mathbb{k})) \tag{6} \\ &= \sum_{0 \leq k < \ell \leq N} \beta_{r_k}^{r_\ell}(\mathbb{k})(f(r_{\ell+1} - r_k) - f(r_{\ell+1} - r_{k+1}) - f(r_\ell - r_k) + f(r_\ell - r_{k+1})). \end{aligned}$$

Now we prove the following inequality:

$$f(r_{\ell+1} - r_k) - f(r_{\ell+1} - r_{k+1}) - f(r_\ell - r_k) + f(r_\ell - r_{k+1}) \geq 0. \tag{7}$$

In addition, if f is strictly convex, the left-hand side is strictly positive. First we prove (7) under the condition of $r_{\ell+1} - r_{k+1} \geq r_\ell - r_k$. In this case,

$$\begin{aligned} & f(r_{\ell+1} - r_k) - f(r_{\ell+1} - r_{k+1}) - f(r_\ell - r_k) + f(r_\ell - r_{k+1}) \\ &= (f(r_{\ell+1} - r_k) - f(r_{\ell+1} - r_{k+1})) - (f(r_\ell - r_k) - f(r_\ell - r_{k+1})) \\ &= f'(\zeta_1)(r_{k+1} - r_k) - f'(\zeta_2)(r_{k+1} - r_k) \\ &= (r_{k+1} - r_k)(f'(\zeta_1) - f'(\zeta_2)), \end{aligned}$$

where $r_{\ell+1} - r_{k+1} \leq \zeta_1 \leq r_{\ell+1} - r_k$ and $r_\ell - r_{k+1} \leq \zeta_2 \leq r_\ell - r_k$. Here, from the assumption of $r_{\ell+1} - r_{k+1} \geq r_\ell - r_k$ and the convexity of f , we have $f'(\zeta_1) - f'(\zeta_2) \geq 0$ and the inequality (7). The strict positivity from strict convexity is trivial. When $r_{\ell+1} - r_{k+1} \leq r_\ell - r_k$, we can prove the inequality in a similar way by exchanging the role of $f(r_{\ell+1} - r_{k+1})$ and $f(r_\ell - r_k)$ in the foregoing.

Since $H_q(X_t; \mathbb{Z})$ is free for any t , $H_q(X_{r_k}; \mathbb{Z}) \rightarrow H_q(X_{r_\ell}; \mathbb{Z})$ has SNF for any k, ℓ . In addition, since $H_{q-1}(X_t; \mathbb{Z})$ is also free for any t , we can apply Theorem 2.4 to have

$$\beta_{r_k}^{r_\ell}(\mathbb{R}) \geq \beta_{r_k}^{r_\ell}(\mathbb{Z}_p) \tag{8}$$

for any p . Furthermore, if $D_q(\mathbb{X}; \mathbb{R}) \neq D_q(\mathbb{X}; \mathbb{Z}_p)$, there exists $k < \ell$ such that

$$\beta_{r_k}^{r_\ell}(\mathbb{R}) > \beta_{r_k}^{r_\ell}(\mathbb{Z}_p) \tag{9}$$

holds. From (6), (7), (8), and (9), we complete the proof of the theorem. □

10 Conclusions

In this paper, we focus on mathematical phenomena concerning the change of the coefficient field in persistent homology. We show that the torsion subgroup of relative homology groups $H_q(X_n, X_m; \mathbb{Z})$ plays an essential role in the phenomena. We also propose an algorithm to judge the independence of the field change. The algorithm is implemented in the software, HomCloud.

Using the algorithm, we conduct experiments which suggest that the probability of persistence diagrams changing as a result of field changes is not zero, but very low for random pointclouds in \mathbb{R}^3 . Thus, we do not need to be particularly concerned about the choice of the field in most practical persistent homology contexts for three-dimensional data if we approach persistence diagrams in statistical terms. To assuage researchers' future concerns about this issue, the torsion condition can be checked by the algorithm.

Of course, where torsion structures are important, such as Klein bottles or Möbius strip, the choice of the coefficient field is important. Based on the results of the numerical experiment on $\tilde{Y}(75)$ and Vietoris–Rips filtrations, we also suggest that the choice of the coefficient field is important for high-dimensional data. In such contexts, further study is required into the torsion on the filtrations.

Further, the results herein suggest that the “difficulty” of computation of $D_q(\mathbb{X}; \mathbb{k})$ depends on the torsion in the filtrations. If the torsion subgroup is zero, $D_q(\mathbb{X}; \mathbb{k})$ for any \mathbb{k} is computable by computing $D_q(\mathbb{X}; \mathbb{k})$ for only one \mathbb{k} , for example, \mathbb{Z}_2 . If not, to compute $D_q(\mathbb{X}; \mathbb{k})$ for many \mathbb{k} is more onerous as explained in Question 1.14. This phenomenon is not dissimilar to a theorem by Dey et al. [11]. Those authors proved that the difficulty of computing a kind of optimization problem on homology algebra depends on the existence of the non-zero torsion subgroup of the relative homology group. Integer programming on homology algebra can be solved by linear programming if the torsion-free condition holds. Integer programming requires much more time than linear programming in the sense of computational complexity theory. Of course, our paper and their paper concern different problems, but the results are similar because of the shared focus on the torsions of relative homology. These results suggest that the existence of non-trivial torsion subgroups in relative homology renders the problems of computational homology is more difficult.

Acknowledgements This work was partially supported by JSPS (Japan Society for the Promotion of Science) KAKENHI Grant Numbers JP 16K17638 and JP 19H00834, JST (Japan Science and Technology Agency) CREST Grant Number JPMJCR15D3, JST PRESTO Grant Number JPMJPR1923, JST-Mirai Program Grant Number JPMJMI18G3, and Osaka Central Advanced Mathematical Institute (MEXT Joint Usage/Research Center on Mathematics and Theoretical Physics JPMXP0619217849). The authors are grateful to the participants at the 2nd JST math workshop on open problems for helpful discussions. The authors also thank Dr. Morozov for the information about the omni-field persistence algorithm.

Funding Open access funding provided the Okayama University.

Open Access This article is licensed under a Creative Commons Attribution 4.0 International License, which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence,

and indicate if changes were made. The images or other third party material in this article are included in the article's Creative Commons licence, unless indicated otherwise in a credit line to the material. If material is not included in the article's Creative Commons licence and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder. To view a copy of this licence, visit <http://creativecommons.org/licenses/by/4.0/>.

References

1. Bauer, U., Kerber, M., Reininghaus, J., Wagner, H.: PHAT – persistent homology algorithms toolbox. *J. Symbol. Comput.* **78**, 76–90 (2017)
2. Boissonnat, J.-D., Maria, C.: Computing persistent homology with various coefficient fields in a single pass. In: 22nd Annual European Symposium on Algorithms (Wrocław 2014). *Lecture Notes in Computer Science*, vol. 8737, pp. 185–196. Springer, Heidelberg (2014)
3. Bubenik, P., Milićević, N.: Homological algebra for persistence modules. *Found. Comput. Math.* **21**(5), 1233–1278 (2021)
4. Carlsson, G.: Topology and data. *Bull. Am. Math. Soc.* **46**(2), 255–308 (2009)
5. Carlsson, G., Ishkhanov, T., de Silva, V., Zomorodian, A.: On the local behavior of spaces of natural images. *Int. J. Comput. Vis.* **76**(1), 1–12 (2008)
6. Chan, J.M., Carlsson, G., Rabadan, R.: Topology of viral evolution. *Proc. Natl. Acad. Sci. U.S.A.* **110**(46), 18566–18571 (2013)
7. Chazal, F., Cohen-Steiner, D., Glisse, M., Guibas, L.J., Oudot, S.Y.: Proximity of persistence modules and their diagrams. In: 25th Annual Symposium on Computational Geometry (Aarhus 2009), pp. 237–246. ACM, New York (2009)
8. Chen, C., Kerber, M.: Persistent homology computation with a twist. In: 27th European Workshop on Computational Geometry (Morschach 2011), pp. 197–200. <https://eurocg11.inf.ethz.ch/docs/Booklet.pdf>
9. Cohen-Steiner, D., Edelsbrunner, H., Harer, J.: Stability of persistence diagrams. *Discrete Comput. Geom.* **37**(1), 103–120 (2007)
10. Cohen-Steiner, D., Edelsbrunner, H., Morozov, D.: Vines and vineyards by updating persistence in linear time. In: 22nd Annual Symposium on Computational Geometry (Sedona 2006), pp. 119–126. ACM, New York (2006)
11. Dey, T.K., Hirani, A.N., Krishnamoorthy, B.: Optimal homologous cycles, total unimodularity, and linear programming. *SIAM J. Comput.* **40**(4), 1026–1044 (2011)
12. Edelsbrunner, H.: Smooth surfaces for multi-scale shape representation. In: *Foundations of Software Technology and Theoretical Computer Science (Bangalore 1995)*. *Lecture Notes in Computer Science*, vol. 1026, pp. 391–412. Springer, Berlin (1995)
13. Edelsbrunner, H., Harer, J.L.: *Computational Topology: An Introduction*. American Mathematical Society, Providence (2010)
14. Edelsbrunner, H., Letscher, D., Zomorodian, A.: Topological persistence and simplification. *Discrete Comput. Geom.* **28**(4), 511–533 (2002)
15. Edelsbrunner, H., Mücke, E.P.: Three-dimensional alpha shapes. *ACM Trans. Graph.* **13**(1), 43–72 (1994)
16. Gakhar, H., Perea, J.A.: Künneth formulae in persistent homology (2019). [arXiv:1910.05656](https://arxiv.org/abs/1910.05656)
17. Hatcher, A.: *Algebraic Topology*. Cambridge University Press, Cambridge (2002)
18. Hiraoka, Y., Nakamura, T., Hirata, A., Escobar, E.G., Matsue, K., Nishiura, Y.: Hierarchical structures of amorphous solids characterized by persistent homology. *Proc. Natl. Acad. Sci. U.S.A.* **113**(26), 7035–7040 (2016)
19. Hiraoka, Y., Shirai, T.: Minimum spanning a cycle and lifetime of persistent homology in the Linial–Meshulam process. *Random Struct. Algorithms* **51**(2), 315–340 (2017)
20. Hu, X., Li, F., Samaras, D., Chen, C.: Topology-preserving deep image segmentation. In: *Advances in Neural Information Processing Systems (NIPS)*, vol. 32. Curran Associates, Red Hook (2019)
21. Ichinomiya, T., Obayashi, I., Hiraoka, Y.: Persistent homology analysis of craze formation. *Phys. Rev. E* **95**, # 012504 (2017)
22. Kahle, M., Lutz, F.H., Newman, A., Parsons, K.: Cohen–Lenstra heuristics for torsion in homology of random complexes. *Exp. Math.* **29**(3), 347–359 (2020)

23. Kimura, M., Obayashi, I., Takeichi, Y., Murao, R., Hiraoka, Y.: Non-empirical identification of trigger sites in heterogeneous processes using persistent homology. *Sci. Rep.* **8**, # 3553 (2018)
24. Knuth, D.E.: *The Art of Computer Programming*. Vol. 3: Sorting and Searching. Addison-Wesley, Reading (1998)
25. Linial, N., Meshulam, R.: Homological connectivity of random 2-complexes. *Combinatorica* **26**(4), 475–487 (2006)
26. Milosavljević, N., Morozov, D., Škraba, P.: Zigzag persistent homology in matrix multiplication time. In: *27th Annual Symposium on Computational Geometry (Paris 2011)*, pp. 216–225. ACM, New York (2011)
27. Otter, N., Porter, M.A., Tillmann, U., Grindrod, P., Harrington, H.A.: A roadmap for the computation of persistent homology. *EPJ Data Sci.* **6**, # 17 (2017)
28. Perea, J.A., Harer, J.: Sliding windows and persistence: an application of topological methods to signal analysis. *Found. Comput. Math.* **15**(3), 799–838 (2015)
29. Polterovich, L., Shelukhin, E., Stojisavljević, V.: Persistence modules with operators in Morse and Floer theory. *Mosc. Math. J.* **17**(4), 757–786 (2017)
30. Saadatfar, M., Takeuchi, H., Robins, V., Francois, N., Hiraoka, Y.: Pore configuration landscape of granular crystallization. *Nat. Commun.* **8**, # 15082 (2017)
31. Spanier, E.H.: *Algebraic Topology*. McGraw-Hill, New York (1966)
32. Storjohann, A.: *Algorithms for Matrix Canonical Forms*. PhD thesis, Swiss Federal Institute of Technology (2000). <https://cs.uwaterloo.ca/~astorjoh/diss2up.pdf>
33. Zomorodian, A., Carlsson, G.: Computing persistent homology. *Discrete Comput. Geom.* **33**(2), 249–274 (2005)

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.