Computing Betti Tables and Minimal Presentations of Zero-Dimensional Persistent Homology

Dmitriy Morozov
International Computer Science Institute, Berkeley, CA, USA
Lawrence Berkeley National Laboratory, Berkeley, CA, USA

Luis Scoccola
Centre de Recherches Mathématiques et Institut des sciences mathématiques, Laboratoire de combinatoire et d'informatique mathématique de l'Université du Québec à, Montréal, Canada
Université de Sherbrooke, Québec, Canada

Abstract

The Betti tables of a multigraded module encode the grades at which there is an algebraic change in the module. Multigraded modules show up in many areas of pure and applied mathematics, and in particular in topological data analysis, where they are known as persistence modules, and where their Betti tables describe the places at which the homology of filtered simplicial complexes changes. Although Betti tables of singly and bigraded modules are already being used in applications of topological data analysis, their computation in the bigraded case (which relies on an algorithm that is cubic in the size of the filtered simplicial complex) is a bottleneck when working with large datasets. We show that, in the special case of 0-dimensional homology (relevant for clustering and graph classification) Betti tables of bigraded modules can be computed in log-linear time. We also consider the problem of computing minimal presentations, and show that minimal presentations of 0-dimensional persistent homology can be computed in quadratic time, regardless of the grading poset.

2012 ACM Subject Classification Mathematics of computing → Algebraic topology; Theory of computation → Computational geometry

Keywords and phrases Multiparameter persistence, Zero-dimensional homology, Minimal presentation, Betti table

Digital Object Identifier 10.4230/LIPIcs.SoCG.2025.69

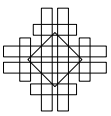
Related Version *Full Version*: `https://arxiv.org/abs/2410.22242` [30]

Funding *Dmitriy Morozov*: National Science Foundation, award DMS-2324632.
Luis Scoccola: National Science Foundation through grants CCF-2006661 and CAREER award DMS-1943758, while at Northeastern University; EPSRC grant "New Approaches to Data Science: Application Driven Topological Data Analysis", EP/R018472/1, while at University of Oxford.

Acknowledgements We thank the anonymous reviewers and the SoCG'25 program committee for helpful comments and for spotting an issue with Definition 24, and its corresponding fix.

1 Introduction

Betti tables and persistence. Betti tables are a classical descriptor of a multigraded modules [18, 29, 34], which encode the grades of the generators in a minimal projective resolution of the module (see, e.g., Figure 1 and Example 15). Informally, one can interpret the Betti tables of a graded module as recording the grades at which there is an algebraic change in the module. Graded modules have applications in a wide variety of areas of pure and applied mathematics, including topological data analysis, and more specifically, persistence theory [33, 6], where they are known as *persistence modules*, and where they are used to describe the varying topology of simplicial complexes and other spaces as they are filtered by one or more real parameters.




Informally, *one-parameter persistence modules* correspond to $\mathbb{Z}$-graded $\Bbbk[x]$-modules, and can thus be classified up to isomorphism effectively. *Multiparameter persistence modules* [11, 6] correspond to $\mathbb{Z}^n$-graded $\Bbbk[x_1, \ldots, x_n]$-modules, and thus do not admit any reasonable classification up to isomorphism (formally, one is dealing with categories of wild representation type [31]; see [11, 2, 4] for manifestations of this phenomenon in persistence theory). For this reason, much of the research in multiparameter persistence is devoted to the study of incomplete descriptors of multiparameter persistence modules. Betti tables (also known as multigraded Betti numbers) provide one of the simplest such descriptors, and various properties of this descriptor from the point of view of persistence theory are well understood, including their effective computation [27, 25, 19, 3], their relationship to discrete Morse theory [22, 1, 21], their optimal transport Lipschitz-continuity with respect to perturbations [32], and their usage in supervised learning [28, 39].

Two-parameter persistent homology. The simplest case beyond the one-parameter case (i.e., the singly graded case) is the two-parameter case (i.e., the bigraded case). Here, one is usually given a finite simplicial complex K together with a function $f : K \longrightarrow \mathbb{R}^2$ mapping the simplices of K to $\mathbb{R}^2$, which is monotonic, i.e., such that $f(\sigma) \leq f(\tau)$ whenever $\sigma \subseteq \tau \in K$ (see Figure 1 for an example). By filtering K using f and taking homology in dimension $i \in \mathbb{N}$ with coefficients in a field $\Bbbk$, one obtains an $\mathbb{R}^2$-graded module, or equivalently, a functor $H_i(K, f; \Bbbk) : \mathbb{R}^2 \longrightarrow \mathbf{Vec}_{\Bbbk}$. Examples include geometric complexes of point clouds filtered by a function on data points, such as a density estimate [8, 12], and graphs representing, say, molecules or networks, filtered by two application-dependent quantities [15]. Applying homology to a bifiltered simplicial complex is justified by the fact that the output is automatically invariant under relabeling, meaning that any operation based on this module, such as computing its Betti tables, will result in a relabeling-invariant descriptor. In this setup, one of the main stability results of [32] implies that, for $f, g : K \longrightarrow \mathbb{R}^2$ any two monotonic functions, we have $\|\mu_f - \mu_g\|_1^{\mathsf{K}} \leq 2 \cdot \|f - g\|_1$, where $\| - \|_1^{\mathsf{K}}$ denotes the Kantorovich–Rubinstein norm between signed measures (also known as the 1-Wasserstein distance), and μ_f and μ_g are the signed measures on $\mathbb{R}^2$ obtained as the alternating sum of Betti tables of $H_i(K, f; \Bbbk)$ and $H_i(K, g; \Bbbk)$, respectively; see [28, Theorem 1] for details. The upshot is that the Betti tables of the homology of bifiltered simplicial complexes form a perturbation-stable, relabeling-invariant descriptor of bifiltered simplicial complexes.

The current standard algorithm for computing the Betti tables of $H_i(K, f; \Bbbk)$ in the two-parameter case is the Lesnick–Wright algorithm [27], which runs in time $O(|K|^3)$. Because of results such as those of [17], one does not expect to find algorithms with better worst-case time complexity than matrix-multiplication time, at least when $i \in \mathbb{N}$ is arbitrary. Current options to speed up practical computations include sparsifying the filtered complex before computing homology [42], as well as computational shortcuts that are known to significantly reduce computational time in practice [25, 3]. Nevertheless, computational cost is still a main bottleneck in real-world applications of persistence, limiting the size of the datasets on which it can be applied.

Zero-dimensional persistent homology. In many applications, persistent homology in dimension zero is all that is required, as it encodes information about the changes in connectivity of filtered simplicial complexes, making it useful for clustering [10, 9, 14, 8, 35, 38] and graph classification [41, 13, 24, 28, 23].

But, if one is only interested in 0-dimensional homology $H_0(K, f; \Bbbk)$, algorithms relying on linear algebra are usually far from the most efficient ones: For example, in the one-parameter case, the Betti tables of the 0-dimensional homology of an $\mathbb{R}$-filtered graph (G, f) can be

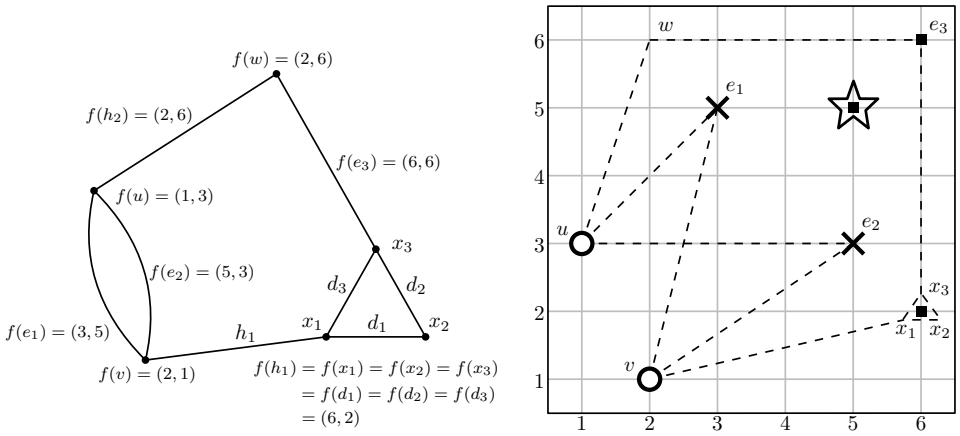


Figure 1 *Left.* A bifiltered graph (G, f) with vertex set $\{u, v, w, x_1, x_2, x_3\}$ and edge set $\{e_1, e_2, e_3, h_1, h_2, d_1, d_2, d_3\}$. *Right.* The bifiltered graph schematically mapped to $\mathbb{R}^2$, together with the Betti tables $\beta_0(H_0(G, f))$ (circles), $\beta_1(H_0(G, f))$ (crosses), $\beta_2(H_0(G, f))$ (stars), and $\beta_0(H_1(G, f))$ (squares).

computed in $O(|G| \log |G|)$ time by first sorting the simplices of G by their f-value, and then doing an ordered pass using a union-find data structure; in the language of barcodes, the Betti tables simply record the endpoints of the barcode, and the barcode can be computed using the elder rule (see [16, pp. 188] or [35, Algorithm 1]).

Contributions. The paper is concerned with the following question: What is the complexity of computing the Betti tables of graphs filtered by posets other than $\mathbb{R}$?

We introduce algorithms for the computation of a minimal set of generators and of a minimal presentation of 0-dimensional persistent homology indexed by an arbitrary poset.

▶ **Theorem A.** *Let $\mathcal{P}$ be any poset, and let (G, f) be a finite $\mathcal{P}$-filtered graph. Algorithm 2 computes the 0th Betti table of $H_0(G, f; \Bbbk)$ in $O(|G|)$ time. Algorithm 3 computes a minimal presentation (and hence the 0th and 1st Betti tables) of $H_0(G, f; \Bbbk)$ in $O(|G|^2)$ time.*

We also introduce a more efficient algorithm specialized to the two-parameter case, which computes all Betti tables, that is, 0th, 1st, and 2nd, by Hilbert's syzygy theorem.

▶ **Theorem B.** *Let (G, f) be a finite $\mathbb{R}^2$-filtered graph. Algorithm 4 computes minimal presentations and the Betti tables of $H_0(G, f; \Bbbk)$ and $H_1(G, f; \Bbbk)$ in $O(|G| \log |G|)$ time.*

Of note is the fact that Betti tables of 0-dimensional two-parameter persistent homology can be computed in log-linear time, as in the one-parameter case.

As another main contribution, we establish a connection between minimal presentations and connectivity properties of filtered graphs (Theorems 23 and 25), which allows us to abstract away the algebraic problem, and focus on a simpler combinatorial problem. The correctness of our algorithms is based on this. Another interesting consequence of these results is that, for arbitrary posets, the 0th and 1st Betti tables of 0-dimensional persistent homology are independent of the field of coefficients, while for the poset $\mathbb{R}^2$, all Betti tables of a filtered graph are independent of the field of coefficients.

Summary of approach and structure of the paper. The main body of the paper has three sections: one on background (Section 2), one on theoretical results (Section 3), and one on algorithms (Section 4). [30, Appendix A] contains proofs.

In order to describe and prove the correctness of our algorithms, we introduce, in the theory section, the notion of a *minimal filtered graph* (Definition 17). Informally, a minimal filtered graph is one whose vertices and edges induce a minimal presentation of its 0-dimensional persistent homology (see Theorem 23). A graph is not minimal if it has some edge that can be either contracted or deleted without changing its 0-dimensional persistent homology (for example, the graph of Figure 1 has both contractible and deletable edges, and is thus not minimal; see Examples 15 and 18). The idea is that, by contracting and deleting edges that do not change 0-dimensional persistent homology, one inevitably ends up with a minimal filtered graph, from which a minimal presentation can be easily extracted. Our main theoretical contributions (Theorems 23 and 25) make this idea precise by giving an explicit minimal presentation of H_0 of any minimal filtered graph, and an explicit minimal resolution of H_0 of any minimal $\mathbb{R}^2$-filtered graph. Our main algorithmic contributions are efficient algorithms for contracting and deleting edges as necessary. Algorithm 4 makes use of a dynamic tree data structure [40], which is the main ingredient that allows us to compute the Betti tables of bifilitered graphs in log-linear time; we give more details in Section 4.

Remark about multi-critical filtrations. The filtrations considered in this paper are 1-*critical*, meaning that each simplex (vertex or edge, in the case of graphs) appears at exactly one grade. In [30, Appendix B], we describe a simple preprocessing step to turn a finite multi-critical filtration by a lattice into a 1-critical filtration with the same 0-dimensional homology. In the two-parameter case, this construction does not change the input complexity, but for other lattices it may increase the input size. Note that the construction does not (necessarily) preserve 1-dimensional homology.

Related work. To the best of our knowledge, the only subcubic algorithm related to Algorithm 4 is that of [8], which, in particular, can be used to compute the Betti tables of H_0 of a function-Rips complex of a finite metric space X in time $O(|X|^2 \log |X|)$. When applied to a function-Rips complex, our Algorithm 4 has the same time complexity; however, our Algorithm 4 applies to arbitrary bifiltered graphs, while function-Rips complexes are very special (and do not include arbitrary filtered graphs). The inner workings of Algorithm 4 are also different from those of [8]: While we rely on dynamic trees, their algorithm relies on a dynamic minimum spanning tree, and, notably, on the fact that, in a function-Rips bifiltration, vertices are filtered exclusively by one of the two filtering functions (so that vertices are linearly ordered in the bifiltration).

The computation of minimal presentations of $\mathbb{R}^n$-filtered complexes is studied in [5]. Since they consider homology in all dimensions, their complexity is significantly worse than the quadratic complexity of Algorithm 3, which only applies to 0-dimensional homology.

Part of the contributions in this paper could be rephrased using the language of discrete Morse theory for multiparameter filtrations [1, 37, 7], specifically, Algorithms 1 and 2 are essentially computing acyclic matchings which respect the filtration. However, this point of view requires extra background, and, to the best of our understanding, cannot be used to describe the more interesting Algorithm 4.

2 Background

As is common in persistence theory, we assume familiarity with very basic notions of category theory, specifically, that of a category and of a functor. We let $\Bbbk$ denote a field, $\mathbf{Vec}_\Bbbk$ denote the category of $\Bbbk$-vector spaces, and $\mathbf{Set}$ denote the category of sets. When the field $\Bbbk$ plays no role, we may denote $\mathbf{Vec}_\Bbbk$ simply by $\mathbf{Vec}$. Proofs can be found in [30, Appendix A.1].

Graphs and filtered graphs. A *graph* $G = (V, E, \partial)$ consists of finite sets V and E, and a function $\partial : E \longrightarrow V \times V$. We refer to the elements of V as *vertices*, typically denoted $v, w, x, y \in V$, and to the elements of E as *edges*, typically denoted $e, d, h \in E$. If $\partial e = (v, w)$, we write $e_0 = v$ and $e_1 = w$. The *size* of a graph G is $|G| = |V| + |E|$.

A *subgraph* of a graph $G = (V, E, \partial)$ is a graph $G' = (V', E', \partial')$, where $V' \subseteq V$, $E' \subseteq E$, and such that $\partial' = \partial|_{E'}$ takes values in $V' \times V' \subseteq V \times V$. If G' is a subgraph of G, we write $G' \subseteq G$.

If $E' \subseteq E$ is a set of edges of G, we let $G \setminus E'$ be the subgraph of G with the same vertices and $E \setminus E'$ as set of edges.

Let $\mathcal{P}$ be a poset. A $\mathcal{P}$-*filtered graph* (G, f^V, f^E) consists of a graph G and functions $f^V : V \longrightarrow \mathcal{P}$ and $f^E : E \longrightarrow \mathcal{P}$ such that $f^V(e_0) \leq f^E(e)$ and $f^V(e_1) \leq f^E(e)$ for all $e \in E$. When there is no risk of confusion, we refer to $\mathcal{P}$-filtered graphs simply as filtered graph, and denote both f^V and f^E by f and the filtered graph (G, f^V, f^E) by (G, f).

If (G, f) is a $\mathcal{P}$-filtered graph and $r \in \mathcal{P}$, we let $(G, f)_r$ be the subgraph of G with vertices $\{v \in V : f(v) \leq r\}$ and edges $\{e \in E : f(e) \leq r\}$.

Persistence modules and persistent sets. Let $\mathcal{P}$ be a poset. A $\mathcal{P}$-*persistence module* is a functor $\mathcal{P} \longrightarrow \mathbf{Vec}$, where $\mathcal{P}$ is the category associated with $\mathcal{P}$. Explicitly, a $\mathcal{P}$-persistence module $M : \mathcal{P} \longrightarrow \mathbf{Vec}$ consists of the following:

- for each $r \in \mathcal{P}$, a vector space $M(r)$;
- for each pair $r \leq s \in \mathcal{P}$, a linear morphism $\varphi^M_{r,s} : M(r) \longrightarrow M(s)$; such that
- for all $r \in \mathcal{P}$, the linear morphism $\varphi^M_{r,r} : M(r) \longrightarrow M(r)$ is the identity;
- for all $r \leq s \leq t \in \mathcal{P}$, we have $\varphi^M_{s,t} \circ \varphi^M_{r,s} = \varphi^M_{r,t} : M(r) \longrightarrow M(t)$.

When there is no risk of confusion, we may refer to a $\mathcal{P}$-persistence module as a persistence module. An *n-parameter persistence module* ($n \geq 1 \in \mathbb{N}$) is an $\mathbb{R}^n$-persistence module.

A *morphism* $g : M \longrightarrow N$ between persistence modules is a natural transformation between functors, that is, a family of linear maps $\{g_r : M(r) \longrightarrow N(r)\}_{r \in \mathcal{P}}$ with the property that $\varphi^N_{r,s} \circ g_r = g_s \circ \varphi^M_{r,s} : M(r) \longrightarrow N(s)$, for all $r \leq s \in \mathcal{P}$. Such a morphism is an *isomorphism* if $g_r : M(r) \longrightarrow N(r)$ is an isomorphism of vector spaces for all $r \in \mathcal{P}$.

If $M, N : \mathcal{P} \longrightarrow \mathbf{Vec}$ are persistence modules, their *direct sum*, denoted $M \oplus N : \mathcal{P} \longrightarrow \mathbf{Vec}$, is the persistence module with $(M \oplus N)(r) \coloneqq M(r) \oplus N(r)$ and with $\varphi^{M \oplus N}_{r,s} \coloneqq \varphi^M_{r,s} \oplus \varphi^N_{r,s} : M(r) \oplus N(r) \longrightarrow M(s) \oplus N(s)$, for all $r \leq s \in \mathcal{P}$.

Similarly, a $\mathcal{P}$-*persistent set* is a functor $\mathcal{P} \longrightarrow \mathbf{Set}$. The concepts of *n-parameter persistent set*, and of *morphism* and *isomorphism* between persistent sets are defined analogously.

Persistent homology and connected components of filtered graphs. If $S \in \mathbf{Set}$, we let $\langle S \rangle_{\Bbbk} \in \mathbf{Vec}$ denote the free vector space generated by S; this defines a functor $\langle - \rangle_{\Bbbk} : \mathbf{Set} \longrightarrow \mathbf{Vec}_{\Bbbk}$. Let $G = (V, E, \partial)$ be a graph. Consider the $\Bbbk$-linear map

$$\begin{aligned} \langle E \rangle_{\Bbbk} &\xrightarrow{d} \langle V \rangle_{\Bbbk} \\ e &\longmapsto e_1 - e_0 \end{aligned} \tag{1}$$

The 0-*dimensional homology* of G, denoted $H_0(G; \Bbbk)$, is the $\Bbbk$-vector space $\operatorname{coker}(d)$, and the 1-*dimensional homology* of G, denoted $H_1(G; \Bbbk)$, is the $\Bbbk$-vector space $\ker(d)$. In particular, every vertex $v \in V$ gives an element $[v] \in H_0(G; \Bbbk)$. When there is no risk of confusion, we omit the field $\Bbbk$ and write $H_i(G)$ instead of $H_i(G; \Bbbk)$.

Homology is functorial, in the following sense. If $G' = (V', E', \partial')$ is a subgraph of G, we have a commutative square

$$\begin{array}{ccc} \langle E' \rangle_{\Bbbk} & \xrightarrow{d'} & \langle V' \rangle_{\Bbbk} \\ \downarrow & & \downarrow \\ \langle E \rangle_{\Bbbk} & \xrightarrow{d} & \langle V \rangle_{\Bbbk} \end{array}$$

induced by the inclusions $V' \subseteq V$ and $E' \subseteq E$. This induces linear maps $H_0(G') \longrightarrow H_0(G)$ and $H_1(G') \longrightarrow H_1(G)$. Moreover, the morphism $H_\bullet(G'') \longrightarrow H_\bullet(G)$ induced by a subgraph $G'' \subseteq G' \subseteq G$ is equal to the composite $H_\bullet(G'') \longrightarrow H_\bullet(G') \longrightarrow H_\bullet(G)$.

If (G, f) is a $\mathcal{P}$-filtered graph, and $i \in \{0, 1\}$, we get a $\mathcal{P}$-persistence module $H_i(G, f) : \mathcal{P} \longrightarrow \mathbf{Vec}$, with $H_i(G, f)(r) = H_i((G, f)_r)$, and with structure morphism $H_i(G, f)(r) \longrightarrow H_i(G, f)(s)$ for $r \leq s \in \mathcal{P}$ induced by the inclusion of graphs $(G, f)_r \subseteq (G, f)_s$.

The set of *connected components* of G, denoted $\pi_0(G)$, is the quotient of V by the equivalence relation $\sim$ where $v \sim w \in V$ if and only if there exists a path in G between v and w. If $v \in V$, we let $[v] \in \pi_0(G)$ denote its connected component, so that $[v] = [w] \in \pi_0(G)$ if and only if v and w belong to the same connected component.

The set of connected components is also functorial with respect to inclusions $G' \subseteq G$, since $[v] = [w] \in \pi_0(G')$ implies $[v] = [w] \in \pi_0(G)$. In particular, if (G, f) is a $\mathcal{P}$-filtered graph, we get a $\mathcal{P}$-persistent set $\pi_0(G, f) : \mathcal{P} \longrightarrow \mathbf{Set}$, with $\pi_0(G, f)(r) = \pi_0((G, f)_r)$, and with the structure morphism $\pi_0(G, f)(r) \longrightarrow \pi_0(G, f)(s)$ for $r \leq s \in \mathcal{P}$ induced by the inclusion of graphs $(G, f)_r \subseteq (G, f)_s$.

The following is straightforward to check.

▶ **Lemma 1.** *If G is a graph, then the map $\langle \pi_0(G) \rangle_{\Bbbk} \longrightarrow H_0(G)$ sending a basis element $[v] \in \pi_0(G)$ to $[v] \in H_0(G)$ is well-defined and an isomorphism of vector spaces. In particular, if (G, f) is a $\mathcal{P}$-filtered graph, composing the persistent set $\pi_0(G, f) : \mathcal{P} \longrightarrow \mathbf{Set}$ with the free vector space functor $\langle - \rangle_{\Bbbk} : \mathbf{Set} \longrightarrow \mathbf{Vec}$ yields a persistence module isomorphic to $H_0(G, f) : \mathcal{P} \longrightarrow \mathbf{Vec}$.* ⌟

Projective persistence modules. Given $r \in \mathcal{P}$, let $\mathsf{P}_r : \mathcal{P} \longrightarrow \mathbf{Vec}$ be the persistence module with $\mathsf{P}_r(s) = \Bbbk$ if $r \leq s$ and $\mathsf{P}_r(s) = 0$ if $r \not\leq s$, with all structure morphisms $\Bbbk \longrightarrow \Bbbk$ being the identity. Equivalently, one can define P_r to be $H_0(\{x\}, \emptyset, \partial, f)$, with $f(x) = r$.

▶ **Notation 2.** *If I is a finite set and $f : I \longrightarrow \mathcal{P}$ is any function, we can consider the direct sum $M = \bigoplus_{i \in I} \mathsf{P}_{f(i)} : \mathcal{P} \longrightarrow \mathbf{Vec}$. When we need to work with elements of such a direct sum, we distinguish summands by writing $M = \bigoplus_{i \in I} \left(\mathsf{P}_{f(i)} \cdot \{i\}\right)$, so that $\left(\mathsf{P}_{f(i)} \cdot \{i\}\right)(r) = 0$ if $r \not\geq f(i)$ and $\left(\mathsf{P}_{f(i)} \cdot \{i\}\right)(r)$ is equal to the free vector space generated by $\{i\}$, for $r \geq f(i)$.*

▶ **Definition 3.** *A persistence module $M : \mathcal{P} \longrightarrow \mathbf{Vec}$ is* projective of finite rank *if there exists a function $\beta^M : \mathcal{P} \longrightarrow \mathbb{N}$ of finite support such that $M \cong \bigoplus_{r \in \mathcal{P}^n} \mathsf{P}_r^{\beta^M(r)}$.*

Note that, drawing inspiration from commutative algebra, projective persistence modules are sometimes also called *free*. The following result justifies the term projective used in Definition 3; see, e.g., [36, Section 3.1] for the usual notion of projective module.

▶ **Lemma 4.** *Let $g : M \longrightarrow N$ be a surjection between $\mathcal{P}$-persistence modules, and let $h : P \longrightarrow N$ with P projective of finite rank. There exists a morphism $h' : P \longrightarrow M$ such that $g \circ h' = h$.*

Resolutions, presentations, and Betti tables. The next result is standard; see, e.g., [26, Proposition 6.24].

▶ **Lemma 5.** *If M is projective of finite rank, then there exists exactly one function (necessarily of finite support) $\beta^M : \mathcal{P} \longrightarrow \mathbb{N}$ such that $M \cong \bigoplus_{r\in\mathcal{P}} \mathsf{P}_r^{\beta^M(r)}$.* ⌟

▶ **Definition 6.** *The* Betti table *of a persistence module $M : \mathcal{P} \longrightarrow \mathbf{Vec}$ that is projective of finite rank is the function $\beta^M : \mathcal{P} \longrightarrow \mathbb{N}$ of Lemma 5.*

The following notation is sometimes convenient.

▶ **Notation 7.** *If $r \in \mathcal{P}$, we let $\delta_r : \mathcal{P} \longrightarrow \mathbb{N}$ be the function defined by $\delta_r(r) = 1$ and $\delta_r(s) = 0$ if $s \neq r$. In particular, $\delta_r = \beta^{\mathsf{P}_r} : \mathcal{P} \longrightarrow \mathbb{N}$.*

▶ **Definition 8.** *Let $M : \mathcal{P} \longrightarrow \mathbf{Vec}$ be a $\mathcal{P}$-persistence module. A* finite projective cover *of M is a surjective morphism $P \longrightarrow M$ where P is projective of finite rank, and $\sum_{r\in\mathcal{P}} \beta^P(r) \in \mathbb{N}$ is minimal.*

▶ **Definition 9.** *Let $M : \mathcal{P} \longrightarrow \mathbf{Vec}$ be a $\mathcal{P}$-persistence module, and let $k \in \mathbb{N}$. A* finite projective k-resolution *(resp.* minimal finite projective k-resolution*) of M, denoted $C_\bullet \longrightarrow M$, is a sequence of morphisms $C_k \xrightarrow{\partial_k} C_{k-1} \xrightarrow{\partial_{k-1}} \cdots \xrightarrow{\partial_2} C_1 \xrightarrow{\partial_1} C_0 \xrightarrow{\partial_0} M$ satisfying*

- *$C_0 \longrightarrow M$ is surjective (resp. a projective cover);*
- *$\partial_i \circ \partial_{i+1} = 0$, for every $0 \leq i \leq k-1$ (so that ∂_{i+1} factors through $\ker \partial_i$);*
- *$C_{i+1} \xrightarrow{\partial_{i+1}} \ker(\partial_i)$ is surjective (resp. a projective cover), for every $0 \leq i \leq k-1$;*
- *C_i is projective of finite rank, for every $0 \leq i \leq k$.*

In particular, a minimal finite projective 0-resolution is simply a projective cover.

▶ **Notation 10.** *Since we only consider finite resolutions, we omit the word "finite" and simply say projective k-resolution. A* (minimal) projective presentation *is a* (minimal) projective 1-resolution.

▶ **Definition 11.** *Let $k \in \mathbb{N}$. A persistence module $M : \mathcal{P} \longrightarrow \mathbf{Vec}$ is* finitely k-resolvable *if it admits a finite projective k-resolution.*

▶ **Definition 12.** *Let $M : \mathcal{P} \longrightarrow \mathbf{Vec}$ be finitely k-resolvable and let $0 \leq i \leq k$. The* ith Betti table *of M is the function $\beta_i^M : \mathcal{P} \longrightarrow \mathbb{N}$ (necessarily of finite support) defined as $\beta_i^M := \beta^{C_i}$, where $C_\bullet \longrightarrow M$ is a minimal k-resolution of M.*

▶ **Notation 13.** *When convenient, we write $\beta_i(M)$ instead of β_i^M for the Betti tables of a persistence module M.*

The Betti tables of M are also sometimes called the *(multigraded) Betti numbers* of M. The Betti tables of M, as defined in Definition 12, are independent of the choice of minimal presentation or resolution, thanks to the following standard result.

▶ **Lemma 14.** *Let $k \in \mathbb{N}$. If $M : \mathcal{P} \longrightarrow \mathbf{Vec}$ is finitely k-resolvable, then it admits a minimal projective k-resolution $C_\bullet \longrightarrow M$. Moreover, any other projective k-resolution $C'_\bullet \longrightarrow M$ has the property that $\beta^{C_i} \leq \beta^{C'_i}$ for all $0 \leq i \leq k$.*

▶ **Example 15.** Consider the graph $G = (V, E, \partial)$ with $V = \{x, y\}$, $E = \{a, b\}$, and $\partial(a) = \partial(b) = (x, y)$. Consider the filtration $f : G \longrightarrow \mathbb{R}^2$ with $f(x) = f(y) = (0, 0)$, $f(a) = (0, 1)$, and $f(b) = (1, 0)$. Then, a minimal resolution of $H_0(G, f)$ is given by

$$0 \longrightarrow \mathsf{P}_{(1,1)} \cdot \{\alpha\} \xrightarrow{\partial_2} \mathsf{P}_{(1,0)} \cdot \{a\} \oplus \mathsf{P}_{(0,1)} \cdot \{b\} \xrightarrow{\partial_1} \mathsf{P}_{(0,0)} \cdot \{x\} \oplus \mathsf{P}_{(0,0)} \cdot \{y\} \xrightarrow{\partial_0} H_0(G, f),$$

where $\partial_0(\{x\}) = [x]$, $\partial_0(\{y\}) = [y]$, $\partial_1(\{a\}) = \partial_1(\{b\}) = \{y\} - \{x\}$, and $\partial_2(\{\alpha\}) = \{a\} - \{b\}$. In particular, $\beta_0(H_0(G,f)) = \delta_{(0,0)} + \delta_{(0,0)}$, $\beta_1(H_0(G,f)) = \delta_{(1,0)} + \delta_{(0,1)}$, and $\beta_2(H_0(G,f)) = \delta_{(1,1)}$. This can be easily checked by hand, but it also follows from Theorems 23 and 25, since (G,f) is a minimal filtered graph, in the sense of Definition 17, which we give in the next section.

3 Theory

We start with a simple, standard result which says that a projective presentation of 0-dimensional homology is given by using the grades of vertices as generators, and the grades of edges as relations. Proofs can be found in [30, Appendix A.2].

▶ **Lemma 16.** *Let $\mathcal{P}$ be a poset, let (G,f) be a $\mathcal{P}$-filtered graph, and consider the following morphism of projective modules*

$$\begin{array}{ccc} \bigoplus_{e\in E} \mathsf{P}_{f(e)} \cdot \{e\} & \xrightarrow{\partial_1^{(G,f)}} & \bigoplus_{v\in V} \mathsf{P}_{f(v)} \cdot \{v\} \\ \{e\} & \longmapsto & \{e_1\} - \{e_0\} \end{array}$$

Then $H_0(G,f) \cong \operatorname{coker}\left(\partial_1^{(G,f)}\right)$ and $H_1(G,f) \cong \ker\left(\partial_1^{(G,f)}\right)$. In particular, $H_0(G,f)$ is a finitely presentable persistence module.

The point of minimal filtered graphs, which we now introduce, is that they make the presentation in Lemma 16 minimal (see Theorem 23).

▶ **Definition 17.** *Let (V,E,∂,f) be a filtered graph and let $e \in E$.*
- *The edge e is* collapsible *if $e_0 \neq e_1$ and $f(e) = f(e_i)$ for some $i \in \{0,1\}$.*
- *The edge e is* deletable *if $[e_0] = [e_1] \in \pi_0(V, E\setminus\{e\}, \partial, f)(f(e))$.*

A filtered graph (G,f) is
- vertex-minimal *if it does not contain any collapsible edges.*
- edge-minimal *if it does not contain any deletable edges.*
- minimal *if it is vertex-minimal and edge-minimal.*

▶ **Example 18.** The graph of Example 15 is minimal, as neither of the two edges is collapsible or deletable. On the other hand, the graph of Figure 1 is not minimal since h_1, h_2, d_1, d_2, and d_3 are collapsible, and e_3 is deletable.

As their name suggests, collapsible and deletable edges can be collapsed or deleted:

▶ **Construction 19.** *Let $(G,f) = (V,E,\partial,f)$ be a filtered graph.*
- *If $e \in E$ is collapsible, let $i \in \{0,1\}$ be such that $f(e) = f(e_i)$. Define the* simple collapse *$G \downarrow e \coloneqq (V\setminus\{e_i\}, E\setminus\{e\}, \partial')$, where $\partial' = (\varphi\times\varphi)\circ\partial$ and $\varphi : V \longrightarrow V$ is given by $\varphi(v) = v$ if $v \neq e_i$ and $\varphi(v) = e_{1-i}$ if $v = e_i$.*
- *If $e \in E$ is any edge, define the* simple deletion *$G \setminus e \coloneqq (V, E\setminus\{e\}, \partial)$.*

We now study the effect of collapsing collapsible edges, and of deleting deletable edges. We start with a useful observation, whose proof is straightforward.

▶ **Lemma 20.** *Let (G,f) be a filtered graph and let e and d be two different edges of G.*
1. *Assume that e is collapsible. If d is not collapsible (resp. deletable) in (G,f), then d is not collapsible (resp. deletable) in $(G\downarrow e, f)$.*
2. *If d is not collapsible (resp. deletable) in (G,f), then d is not collapsible (resp. deletable) in $(G\setminus e, f)$.* ⌟

▶ **Lemma 21.** *If $(G, f) = (V, E, \partial, f)$ is a filtered graph and $e \in E$ is collapsible, then $H_i(G, f) \cong H_i(G \downarrow e, f)$ for $i \in \{0, 1\}$.*

▶ **Lemma 22.** *If $(G, f) = (V, E, \partial, f)$ is a filtered graph and $e \in E$ is deletable, then $H_0(G, f) \cong H_0(G \setminus e, f)$ and $H_1(G, f) \cong H_1(G \setminus e, f) \oplus \mathsf{P}_{f(e)}$.*

Next is a construction of a minimal presentation of H_0 of a minimal filtered graph.

▶ **Theorem 23.** *Let $\mathcal{P}$ be a poset, let (G, f) is a $\mathcal{P}$-filtered graph, and let $\partial_1^{(G,f)} : C_1 \longrightarrow C_0$ be the morphism of Lemma 16.*

1. *If (G, f) is vertex-minimal, then the map $C_0 \longrightarrow \operatorname{coker}(\partial_1^{(G,f)})$ is a projective cover. In this case, $\beta_0\big(H_0(G, f)\big) = \sum_{v \in V} \delta_{f(v)}$.*
2. *If (G, f) is a minimal filtered graph, then $C_1 \xrightarrow{\partial_1^{(G,f)}} C_0 \longrightarrow \operatorname{coker}(\partial_1^{(G,f)})$ is a minimal presentation. In this case, we also have $\beta_1\big(H_0(G, f)\big) = \sum_{e \in E} \delta_{f(e)}$.*

We now focus on the case of $\mathbb{R}^2$-filtered graphs. If (G, f) is an $\mathbb{R}^2$-filtered graph, let $f_{\mathbf{x}}, f_{\mathbf{y}} : G \longrightarrow \mathbb{R}$ denote the first and second coordinates of f, respectively.

▶ **Definition 24.** *Let (V, E, ∂, f) be a minimal $\mathbb{R}^2$-filtered graph, and let $\prec$ be a total order on the set of edges E that refines the lexicographic order induced by f (i.e., $e \prec e'$ implies that $f_{\mathbf{x}}(e) < f_{\mathbf{x}}(e')$ or $f_{\mathbf{x}}(e) = f_{\mathbf{x}}(e')$ and $f_{\mathbf{y}}(e) \leq f_{\mathbf{y}}(e')$). An edge $d \in E$ is* cycle-creating *with respect to $\prec$ if $[d_0] = [d_1] \in \pi_0(V, \{e \in E : e \prec d\}, \partial)$.*

Thus, $d \in E$ is cycle-creating if there exists a list of edges $e^1, \ldots, e^k \in E$ with $e^i \prec d$ for $1 \leq i \leq k$, and a list of signs $s^1, \ldots, s^k \in \{+, -\}$, such that $s^1 e^1, \ldots, s^k e^k$ is a directed path from d_0 to d_1. Such a path $w = (s^\bullet, e^\bullet)$ is called a *witness* for d, and we denote $f(w) \coloneqq \big(f_{\mathbf{x}}(d), \max_{1 \leq i \leq k} f_{\mathbf{y}}(e^i)\big)$. A *minimal witness* w of a cycle-creating edge d is one for which $f_{\mathbf{y}}(w) \in \mathbb{R}$ is as small as possible.

Since $\prec$ refines the lexicographic order, we have $f_{\mathbf{x}}(e^i) \leq f_{\mathbf{x}}(d)$ for all i; thus, if $\max_{1 \leq i \leq k} f_{\mathbf{y}}(e^i) \leq f_{\mathbf{y}}(d)$, we would have that $[d_0] = [d_1] \in \pi_0(V, E \setminus \{d\}, \partial, f)(f(d))$, which contradicts the fact that the filtered graph is minimal. This means that if w is a witness for d, then $\max_{1 \leq i \leq k} f_{\mathbf{y}}(e^i) > f_{\mathbf{y}}(d)$, and thus $f(d) < f(w)$.

▶ **Theorem 25.** *Let $(G, f) = (V, E, \partial, f)$ be a minimal $\mathbb{R}^2$-filtered graph, and $\prec$ a total order on E refining the lexicographic order. For each $d \in E$ cycle-creating, let $w_d = (s_d^\bullet, e_d^\bullet)$ be a minimal witness wrt to $\prec$. The kernel of the morphism $\partial_1^{(G,f)}$ of Lemma 16 is given by:*

$$\begin{array}{ccc} \bigoplus_{\substack{d \in E \\ \text{cycle-creating}}} \mathsf{P}_{f(w_d)} \cdot \{d\} & \xrightarrow{\kappa^{(G,f)}} & \bigoplus_{e \in E} \mathsf{P}_{f(e)} \cdot \{e\} \\ \{d\} & \longmapsto & \{d\} - \big(s_d^1 \{e_d^1\} + \cdots + s_d^k \{e_d^k\}\big). \end{array}$$

It follows that

- $\beta_0(H_0(G, f)) = \sum_{v \in V} \delta_{f(v)}$;
- $\beta_1(H_0(G, f)) = \sum_{e \in E} \delta_{f(e)}$;
- $\beta_0(H_1(G, f)) = \beta_2(H_0(G, f)) = \sum_{d \in E, \text{cycle-creating}} \delta_{f(w_d)}$;
- $\beta_i(H_0(G, f)) = 0$ *for* $i \geq 3$, *and* $\beta_i(H_1(G, f)) = 0$ *for* $i \geq 1$.

4 Algorithms

Outline. We first introduce some technical notions and overview the algorithms.

Two $\mathcal{P}$-filtered graph (G, f) and (G', f') are *homology-equivalent* (over $\Bbbk$) if $H_0(G, f; \Bbbk) \cong H_0(G', f'; \Bbbk)$ and $H_1(G, f; \Bbbk) \cong H_1(G', f'; \Bbbk)$.

Algorithm 2 reduces the input filtered graph to a vertex-minimal filtered graph by collapsing edges; it relies on Algorithm 1, which collapses local collapsible edges. A *local collapsible edge* of (G, f) is an edge e such that $e_0 \neq e_1$ and $f(e) = f(e_0) = f(e_1)$. Algorithm 2 then identifies *minimal vertices*, that is, vertices with no adjacent collapsible edge decreasing the grade, and runs a depth-first search from each of these to build and collapse a tree of collapsible edges.

Algorithm 3 first calls Algorithm 2 to perform all collapses and then deletes deletable edges until there are no more deletable edges. It then builds the presentation of Lemma 16.

Algorithm 4 first calls Algorithm 2 to perform all collapses, and then goes through all edges, with respect to the lex order on their grades, and identifies deletable edges, non-deletable edges, and cycle-creating edges.

Dynamic dendrograms. Since this is used in Algorithm 4, we now introduce the dynamic dendrogram data structure, which, informally, represents a dendrogram where elements merge as time goes from $-\infty$ to ∞, and is dynamic in that one is allowed to change the dendrogram by making two elements merge earlier.

▶ **Definition 26.** *Let $(G, f) = (V, E, \partial, f)$ be a $[-\infty, \infty)$-filtered graph. A* dynamic dendrogram *D for (G, f) is a data structure supporting the following operations:*

- *If $v, w \in V$, the operation $D.\mathsf{time\_of\_merge}(v, w)$ returns the smallest $r \in [-\infty, \infty)$ such that $[v] = [w] \in \pi_0(G, f)(r)$, or ∞ if $[v] \neq [w] \in \pi_0(G, f)(r)$ for all $r \in [-\infty, \infty)$.*
- *If $v, w \in V$ and $t \in [-\infty, \infty)$, the operation $D.\mathsf{merge\_at\_time}(v, w, t)$ modifies the dendrogram so that it is a dynamic dendrogram for $(V, E \sqcup \{e\}, \partial', f')$, with $f'|_E = f$, $\partial'|_E = \partial$, $f'(e) = t$, and $\partial'(e) = (v, w)$.*

A dynamic dendrogram D can easily and efficiently be implemented using a *mergeable tree* T in the sense of [20]. These trees store heap-ordered forests, i.e., a collection of rooted labeled trees, where the labels decrease (or in our case increase) along paths to the root. The data structures support a wealth of operations for dynamic updates. We need only three of them: $\mathsf{insert}(v, t)$ inserts node v with label t into the forest; $\mathsf{nca}(v, w)$ finds the nearest common ancestor of two nodes v and w; $\mathsf{merge}(v, w)$ merges the paths of v and w to their root(s) while preserving the heap order. We use these operations to implement dynamic dendrograms as follows:

- To implement $D.\mathsf{time\_of\_merge}(v, w)$, return the label of $T.\mathsf{nca}(v, w)$.
- To implement $D.\mathsf{merge\_at\_time}(v, w, t)$, let h be a new vertex not already in the mergeable tree T, do $T.\mathsf{insert}(h, t)$, then $T.\mathsf{merge}(v, h)$, and $T.\mathsf{merge}(w, h)$.

Presentation matrices. In the algorithms, to represent a Betti table $\beta_i(M)$ we use a list of elements of the indexing poset. If we have represented the 0th and 1st Betti tables of M with lists β_0 and β_1, we represent a minimal presentation for M with a sparse matrix using coordinate format, that is, with a list of triples (i, j, v) representing the fact that v is the entry at row i and column j, and where i represents the ith element of β_0 and j represents the jth element of β_1.

Algorithm 1 Collapse local collapsible edges.

Input: Filtered graph $(G, f) = (V, E, \partial, f)$
Output: Filtered graph (G', f) homology-equivalent to (G, f) and without local collapsible edges

Initialize dictionary φ with identity map, $\varphi[v] = v$ for $v \in V$
Initialize empty set visited
Initialize $E' \leftarrow E$
for $v \in V$ **do** ▷ Run depth-first search from v on local edges
  Initialize stack with $(v, \emptyset, v)$
  while stack *is not empty* **do**
    $(v, e, u) \leftarrow$ stack.**pop**()
    if $u \notin$ visited **then**
      add u to visited
      if $e \neq \emptyset$ **then**
        $\varphi[u] \leftarrow v$
        remove e from E'
      for $e \in E$ *with* $\{e_0, e_1\} = \{u, x\}$ *and* $f(e) = f(x) = f(u)$ **do** ▷ local collapsible
        push (v, e, x) onto stack
$V' \leftarrow \{v \in V : \varphi[v] = v\}$
return $(G', f) = (V', E', (\varphi \times \varphi) \circ \partial, f)$

Algorithm 2 Collapse to vertex-minimal filtered graph.

Input: Filtered graph $(G, f) = (V, E, \partial, f)$
Output: Vertex-minimal filtered graph (G', f) homology-equivalent to (G, f), and $\beta_0(H_0(G, f))$

$(G_i, f) = (V_i, E_i, \partial_i, f) \leftarrow$ Algorithm 1(G, f)
Initialize dictionary φ with identity map, $\varphi[v] = v$ for $v \in V_i$
Initialize empty set visited
Initialize $E' \leftarrow E_i$
for v *in* V_i **do** ▷ Run depth-first search from minimal vertices
  if $\exists$ *edge* $e \in E_i$ *with* $\{e_0, e_1\} = \{u, v\}$ *and* $f(e) = f(v) > f(u)$ **then** ▷ v is not minimal
    continue
  Initialize stack with $(v, \emptyset, v)$
  while stack *is not empty* **do**
    $v, e, u \leftarrow$ stack.**pop**()
    if $u \notin$ visited **then**
      add u to visited
      if $e \neq \emptyset$ **then**
        $\varphi[u] \leftarrow v$
        remove e from E'
      for *every edge* $e \in E_i$ *with* $\{e_0, e_1\} = \{u, x\}$ *with* $f(e) = f(x) > f(u)$ **do**
        push (v, e, x) onto stack
$V' \leftarrow \{v \in V_i : \varphi[v] = v\}$
return $(G', f) = (V', E', (\varphi \times \varphi) \circ \partial_i, f)$ and $\beta_0 = [f(v) : v \in V']$

Algorithm 3 Minimal presentation of $\mathcal{P}$-filtered graph.

Input: Filtered graph $(G, f) = (V, E, \partial, f)$
Output: Minimal presentation of $H_0(G, f; \Bbbk)$

Initialize β_0, β_1 with empty lists
Initialize empty sparse matrix $\mathcal{M}$ and empty dictionary row_idx
$(V', E', \partial', f) \leftarrow$ Algorithm 2(G, f)
for $e \in E'$ **do** ▷ Check each edge, and delete it if it is deletable
  Define set of vertices $V_e \leftarrow \{v \in V' : f(v) \leq f(e)\}$
  Define set of edges $E_e \leftarrow \{d \in E' : f(d) \leq f(e)\} \setminus e$
  Run breadth-first search on (V_e, E_e, ∂') starting from e_0
  if e_1 *is reachable from* e_0 **then** ▷ $[e_0] = [e_1] \in \pi_0(V_e, E_e, \partial')$ so e is deletable
    $E' \leftarrow E' \setminus \{e\}$
for $v \in V'$ **do**
  β_0.append$(f(v))$
  row_idx$[v] \leftarrow |\beta_0|$
for $e \in E'$ **do** ▷ The morphism $\partial_1^{(V', E', \partial')}$ of Lemma 16
  β_1.append$(f(e))$
  $\mathcal{M}$.append(row_idx$[e_0]$, $|\beta_1|$, -1)
  $\mathcal{M}$.append(row_idx$[e_1]$, $|\beta_1|$, 1)
return $\beta_0, \beta_1, \mathcal{M}$

Algorithm 4 Betti tables and minimal presentation of $\mathbb{R}^2$-filtered graph.

Input: Filtered graph $(G, f) = (V, E, \partial, f)$
Output: Betti tables $\beta_0, \beta_1, \beta_2$ and minimal presentation of $H_0(G, f; \Bbbk)$, and $\beta_0^1 \coloneqq \beta_0(H_1(G, f; \Bbbk))$

Initialize $\beta_0, \beta_1, \beta_2, \beta_0^1$ with empty lists
Initialize empty sparse matrix $\mathcal{M}$ and empty dictionary row_idx
$(V', E', \partial', f) \leftarrow$ Algorithm 2(G, f)
Let D be a dynamic dendrogram on $(V', \emptyset, \partial, g)$, with $g(v) = -\infty$ for all $v \in V'$
for $(x, y) \in \mathbb{R}^2$ *s.t.* $f^{-1}(x, y) \neq \emptyset$ *in lex order* **do** ▷ Visit grades lexicographically
  for $v \in V'$ *with* $f(v) = (x, y)$ **do** ▷ All vertices belong to the projective cover
    β_0.append((x, y))
    row_idx$[v] \leftarrow |\beta_0|$
  for $e \in E'$ *with* $f(e) = (x, y)$ **do**
    $s \leftarrow D$.time_of_merge(e_0, e_1)
    if $s \leq y$ **then** ▷ The edge is deletable, so it only affects H_1
      β_0^1.append((x, y))
    else ▷ Edge is not deletable, so belongs to relations in resolution
      D.merge_at_time(e_0, e_1, y)
      β_1.append((x, y))
      $\mathcal{M}$.append(row_idx$[e_0]$, $|\beta_1|$, -1)
      $\mathcal{M}$.append(row_idx$[e_1]$, $|\beta_1|$, 1)
      if $s < \infty$ **then** ▷ The edge is cycle-creating
        β_2.append((x, s))
        β_0^1.append((x, s))
return $\beta_0, \beta_1, \beta_2, \beta_0^1, \mathcal{M}$

Complexity and correctness. We conclude the paper by using the theoretical results of Section 3 to prove the main results in the introduction. Proofs for the results in this section are in [30, Appendix A.3].

We start by with a convenient lemma, which gives conditions under which one can collapse an entire subgraph and produce a homology-equivalent filtered graph. The following definition describes the type of subgraph that can be collapsed.

▶ **Definition 27.** *Let $(G, f) = (V, E, \partial, f)$ be a filtered graph. A subset $E' \subseteq E$ is a* monotonic forest *of (G, f) if:*

1. *The subgraph of G spanned by the edges in E' is a forest.*
2. *For every vertex v in the forest, there exists at most one edge $e \in E'$ such that $\{e_0, e_1\} = \{v, w\}$ and $f(w) < f(v)$.*

▶ **Lemma 28.** *Let $(G, f) = (V, E, \partial, f)$ be a filtered graph, and let $E' \subseteq E$ be a set of collapsible edges, which forms a monotonic forest. If $e \in E'$, then all the edges in $E' \setminus \{e\}$ are collapsible in $(G \downarrow e, f)$. In particular, the whole forest E' can be collapsed (in any order) to obtain a filtered graph that is homology-equivalent to (G, f).*

▶ **Lemma 29.** *Let (G, f) be a $\mathcal{P}$-filtered graph. Algorithm 1 outputs a filtered graph homology-equivalent to (G, f) and without local collapsible edges in time $O(|G|)$.*

▶ **Proposition 30.** *Let (G, f) be a $\mathcal{P}$-filtered graph. Algorithm 2 outputs a vertex-minimal filtered graph homology-equivalent to (G, f) and $\beta_0(H_0(G, f))$ in $O(|G|)$ time.*

▶ **Proposition 31.** *Let (G, f) be a finite $\mathcal{P}$-filtered graph. Algorithm 3 outputs a minimal presentation of $H_0(G, f)$ in $O(|G|^2)$ time.*

With these results, we prove Theorem A and Theorem B in [30, Appendix A.4].

References

1 Madjid Allili, Tomasz Kaczynski, and Claudia Landi. Reducing complexes in multidimensional persistent homology theory. *J. Symbolic Comput.*, 78:61–75, 2017. doi:10.1016/j.jsc.2015.11.020.

2 Ulrich Bauer, Magnus B. Botnan, Steffen Oppermann, and Johan Steen. Cotorsion torsion triples and the representation theory of filtered hierarchical clustering. *Adv. Math.*, 369:107171, 51, 2020. doi:10.1016/j.aim.2020.107171.

3 Ulrich Bauer, Fabian Lenzen, and Michael Lesnick. Efficient two-parameter persistence computation via cohomology. In *39th International Symposium on Computational Geometry*, volume 258 of *LIPIcs. Leibniz Int. Proc. Inform.*, pages Art. No. 15, 17. Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2023. doi:10.4230/LIPIcs.SoCG.2023.15.

4 Ulrich Bauer and Luis Scoccola. Multi-parameter persistence modules are generically indecomposable. *International Mathematics Research Notices*, 2025(5):rnaf034, February 2025. doi:10.1093/imrn/rnaf034.

5 Matías Bender, Oliver Gäfvert, and Michael Lesnick. Efficient computation of multiparameter persistence, (in preparation).

6 Magnus Bakke Botnan and Michael Lesnick. An introduction to multiparameter persistence. In *Representations of algebras and related structures*, EMS Ser. Congr. Rep., pages 77–150. Eur. Math. Soc., Zürich, [2023] ©2023.

7 Guillaume Brouillette, Madjid Allili, and Tomasz Kaczynski. Multiparameter discrete morse theory. *Journal of Applied and Computational Topology*, pages 1–42, 2024.

8 Chen Cai, Woojin Kim, Facundo Mémoli, and Yusu Wang. Elder-rule-staircodes for augmented metric spaces. *SIAM J. Appl. Algebra Geom.*, 5(3):417–454, 2021. doi:10.1137/20M1353605.

9 Gunnar Carlsson and Facundo Mémoli. Multiparameter hierarchical clustering methods. In *Classification as a tool for research*, Stud. Classification Data Anal. Knowledge Organ., pages 63–70. Springer, Berlin, 2010. doi:10.1007/978-3-642-10745-0_6.

10 Gunnar Carlsson and Facundo Mémoli. Classifying clustering schemes. *Found. Comput. Math.*, 13(2):221–252, 2013. doi:10.1007/s10208-012-9141-9.

11 Gunnar Carlsson and Afra Zomorodian. The theory of multidimensional persistence. In *Computational geometry (SCG'07)*, pages 184–193. ACM, New York, 2007. doi:10.1145/1247069.1247105.

12 Mathieu Carrière and Andrew J. Blumberg. Multiparameter persistence images for topological machine learning. In *Proceedings of the 34th International Conference on Neural Information Processing Systems*, NIPS '20, Red Hook, NY, USA, 2020. Curran Associates Inc.

13 Mathieu Carriere, Frederic Chazal, Yuichi Ike, Theo Lacombe, Martin Royer, and Yuhei Umeda. Perslay: A neural network layer for persistence diagrams and new graph topological signatures. In Silvia Chiappa and Roberto Calandra, editors, *Proceedings of the Twenty Third International Conference on Artificial Intelligence and Statistics*, volume 108 of *Proceedings of Machine Learning Research*, pages 2786–2796. PMLR, 26–28 August 2020. URL: https://proceedings.mlr.press/v108/carriere20a.html.

14 Frédéric Chazal, Leonidas J. Guibas, Steve Y. Oudot, and Primoz Skraba. Persistence-based clustering in Riemannian manifolds. *J. ACM*, 60(6):Art. 41, 38, 2013. doi:10.1145/2535927.

15 Andac Demir, Baris Coskunuzer, Ignacio Segovia-Dominguez, Yuzhou Chen, Yulia Gel, and Bulent Kiziltan. Todd: topological compound fingerprinting in computer-aided drug discovery. In *Proceedings of the 36th International Conference on Neural Information Processing Systems*, NIPS '22, Red Hook, NY, USA, 2024. Curran Associates Inc.

16 Herbert Edelsbrunner and John L. Harer. *Computational topology*. American Mathematical Society, Providence, RI, 2010. An introduction. doi:10.1090/mbk/069.

17 Herbert Edelsbrunner and Salman Parsa. On the computational complexity of Betti numbers: reductions from matrix rank. *Proceedings of the twenty-fifth annual ACM-SIAM symposium on discrete algorithms*, pages 152–160, 2014. doi:10.1137/1.9781611973402.11.

18 David Eisenbud. *Commutative algebra*, volume 150 of *Graduate Texts in Mathematics*. Springer-Verlag, New York, 1995. With a view toward algebraic geometry. doi:10.1007/978-1-4612-5350-1.

19 Ulderico Fugacci, Michael Kerber, and Alexander Rolle. Compression for 2-parameter persistent homology. *Comput. Geom.*, 109:Paper No. 101940, 28, 2023. doi:10.1016/j.comgeo.2022.101940.

20 Loukas Georgiadis, Haim Kaplan, Nira Shafrir, Robert E. Tarjan, and Renato F. Werneck. Data structures for mergeable trees. *ACM Trans. Algorithms*, 7(2), March 2011. doi:10.1145/1921659.1921660.

21 Andrea Guidolin and Claudia Landi. Morse inequalities for the Koszul complex of multipersistence. *J. Pure Appl. Algebra*, 227(7):Paper No. 107319, 26, 2023. doi:10.1016/j.jpaa.2023.107319.

22 Andrea Guidolin and Claudia Landi. On the support of betti tables of multiparameter persistent homology modules. *arXiv preprint arXiv:2303.05294*, 2023.

23 Olympio Hacquard and Vadim Lebovici. Euler characteristic tools for topological data analysis. *Journal of Machine Learning Research*, 25(240):1–39, 2024. URL: http://jmlr.org/papers/v25/23-0353.html.

24 Christoph Hofer, Florian Graf, Bastian Rieck, Marc Niethammer, and Roland Kwitt. Graph filtration learning. In Hal Daumé III and Aarti Singh, editors, *Proceedings of the 37th International Conference on Machine Learning*, volume 119 of *Proceedings of Machine Learning Research*, pages 4314–4323. PMLR, 13–18 July 2020. URL: https://proceedings.mlr.press/v119/hofer20b.html.

25 Michael Kerber and Alexander Rolle. Fast minimal presentations of bi-graded persistence modules. *Proceedings of the Symposium on Algorithm Engineering and Experiments (ALENEX)*, pages 207–220, 2021. doi:10.1137/1.9781611976472.16.

26 Michael Lesnick. Notes on multiparameter persistence (for amat 840), 2023.

27 Michael Lesnick and Matthew Wright. Computing minimal presentations and bigraded Betti numbers of 2-parameter persistent homology. *SIAM J. Appl. Algebra Geom.*, 6(2):267–298, 2022. doi:10.1137/20M1388425.

28 David Loiseaux, Luis Scoccola, Mathieu Carrière, Magnus Bakke Botnan, and Steve Oudot. Stable vectorization of multiparameter persistent homology using signed barcodes as measures. *Advances in Neural Information Processing Systems*, 36, 2024.

29 Ezra Miller and Bernd Sturmfels. *Combinatorial commutative algebra*, volume 227 of *Graduate Texts in Mathematics*. Springer-Verlag, New York, 2005.

30 Dmitriy Morozov and Luis Scoccola. Computing betti tables and minimal presentations of zero-dimensional persistent homology, 2024. doi:10.48550/arXiv.2410.22242.

31 L. A. Nazarova. Partially ordered sets of infinite type. *Izv. Akad. Nauk SSSR Ser. Mat.*, 39(5):963–991, 1219, 1975.

32 Steve Oudot and Luis Scoccola. On the Stability of Multigraded Betti Numbers and Hilbert Functions. *SIAM J. Appl. Algebra Geom.*, 8(1):54–88, 2024. doi:10.1137/22M1489150.

33 Steve Y. Oudot. *Persistence theory: from quiver representations to data analysis*, volume 209 of *Mathematical Surveys and Monographs*. American Mathematical Society, Providence, RI, 2015. doi:10.1090/surv/209.

34 Irena Peeva. *Graded syzygies*, volume 14 of *Algebra and Applications*. Springer-Verlag London, Ltd., London, 2011. doi:10.1007/978-0-85729-177-6.

35 Alexander Rolle and Luis Scoccola. Stable and consistent density-based clustering via multiparameter persistence. *Journal of Machine Learning Research*, 25(258):1–74, 2024. URL: http://jmlr.org/papers/v25/21-1185.html.

36 Joseph J. Rotman. *An introduction to homological algebra*. Universitext. Springer, New York, second edition, 2009. doi:10.1007/b98977.

37 Sara Scaramuccia, Federico Iuricich, Leila De Floriani, and Claudia Landi. Computing multiparameter persistent homology through a discrete morse-based approach. *Computational Geometry*, 89:101623, 2020. doi:10.1016/j.comgeo.2020.101623.

38 Luis Scoccola and Alexander Rolle. Persistable: persistent and stable clustering. *Journal of Open Source Software*, 8(83):5022, 2023. doi:10.21105/joss.05022.

39 Luis Scoccola, Siddharth Setlur, David Loiseaux, Mathieu Carrière, and Steve Oudot. Differentiability and optimization of multiparameter persistent homology. In *Forty-first International Conference on Machine Learning*, 2024. URL: https://openreview.net/forum?id=ixdfvnO0uy.

40 Daniel D. Sleator and Robert Endre Tarjan. A data structure for dynamic trees. *J. Comput. System Sci.*, 26(3):362–391, 1983. doi:10.1016/0022-0000(83)90006-5.

41 Qi Zhao and Yusu Wang. *Learning metrics for persistence-based summaries and applications for graph classification*, pages 9859 – 9870. Curran Associates Inc., Red Hook, NY, USA, 2019.

42 Ángel Javier Alonso, Michael Kerber, and Siddharth Pritam. Filtration-domination in bifiltered graphs. *2023 Proceedings of the Symposium on Algorithm Engineering and Experiments (ALENEX)*, pages 27–38, 2023. doi:10.1137/1.9781611977561.ch3.