POLITECNICO
DI TORINO
UNIVERSITÀ
DEGLI STUDI
DI TORINO

Doctoral Dissertation
Doctoral Program in Pure and Applied Mathematics (34^{th}cycle)

Persistent Homology, from localizing cycles to generating cycles

By

Marco Guerra

******

Supervisor:
Prof. Francesco Vaccarino, Supervisor

Università di Torino - Politecnico di Torino
2022

Declaration

I hereby declare that the contents and organization of this dissertation constitute my own original work and does not compromise in any way the rights of third parties, including those relating to the security of personal data.

Marco Guerra
2022

* This dissertation is presented in partial fulfillment of the requirements for the degree of ***Philosophiæ Diploma*** **(PhD degree)** in **Pure and Applied Mathematics**.

Abstract

This thesis work is centred around the subject of Topological Data Analysis, a modern line of study that combines tools from algebraic and computational topology. After the introduction, a background section introduces the basic concepts in category theory, abstract algebra and algebraic topology to provide footing for the following, which is split into two main chapters.
The first chapter generally deals with the topic of representative cycles for homology; this responds to the intuitive idea of associating a shape to a homology class. Several approaches exist to address the problem. In this chapter we present results about minimal bases, that were employed to construct the so-called minimal scaffold, as well as mixed approaches such as the homologically-persistent skeleton, and we report preliminary ideas about the usage of Alexander duality to obtain canonical representatives.
The second chapter is devoted to the exposition of work that was carried out regarding the topic of the decomposition of persistence modules. In particular, we define a notion of interval basis, a choice of generators whose interval modules generate the direct-sum decomposition of the structure theorem, and which can be computed in parallel. We provide several algorithms via different approaches for the computation of an interval basis, and conclude the chapter with an exposition of the parallel construction of persistent homology from a sequence of chain maps, in particular via the Hodge Laplacian.
A brief section about future avenues of work concludes the thesis.

Contents

List of Figures

Chapter 1

Introduction

Topological data analysis is a recent branch of mathematics that stemmed from the intuition that classical tools in algebraic topology could provide valuable descriptors for the analysis of data.

Its origins can be traced back to the late 90's with works of Frosini, Edelsbrunner and Robins, and it has witnessed a dramatic expansion within the last twenty years, as it provides a mathematically rigorous and powerful set of tools. The key idea of persistent homology, without question the foremost concept in TDA, is that of studying the maps induced between homology spaces along a suitable filtered topological space. The dynamic, evolutive, nature of this process is what made persistent homology stand out as an unmatched instrument for shape description and comparison. Furthermore, the broad scope of algebraic topology in terms of what notions of *space* can be endowed with a description of their *shape* has made TDA extremely flexible as to what types of data can be processed. The well-developed theory of simplicial complexes is naturally suited for the analysis of discrete objects, and real data is, in the vast majority of cases, a discrete set of measurements. At the same time, under very reasonable hypotheses, the many different theories of shape agree throughout a broad range of descriptions of their underlying space.

Key steps for the development of the theory were undoubtedly the formulation of the algebraic characterization of persistence modules, and the stability results that soon followed. A vast and mathematically intriguing theory of *multiparameter* persistent homology was formulated since the early days, and is currently the subject of intense inquiry. A large body of work was devoted to the computational aspects of TDA, bringing about numerous libraries and software packages, that make it into a reliable

and nearly off-the-shelf methodology for the applications.
In its maturity years, TDA has been given a rigorous footing within category theory and algebra, and its concepts have gained mathematical status per se, even being applied in pure mathematics. It was soon clear that methods needed to be developed to interface TDA with classical machine inference tools. The large thread of vectorization methods responded to precisely that need, and indeed TDA began to find use in a host of different applicative context. Nowadays, it interfaces with subjects as diverse as neuroscience and random graphs, material science and cosmology, genomics and complex networks.
The current research on TDA is active in essentially each one of the branches we have mentioned. Multipersistence, due to its wilder mathematical nature, is being tackled from a range of points of view. The topology of random phenomena is an involved and interesting line of research. Many efforts are in place to interface computational topology with statistical inference, in terms of not just using topology as a feature but of actually inferring about the topology. Along the same line, in the last 4 or 5 years there was a surge of works exploring the many ways in which TDA can be interfaced with machine learning, as a method of explicating how obscure neural networks black boxes work, or as a regularization strategy, and even as a loss function per se.

The purpose of this thesis work is to narrate some of the avenues that I pursued during the course of my PhD. It will contain material at different stages of development, while many topics have, for different reasons, been left out.
The first chapter (actually chapter 2) is devoted to a brief survey of the mathematical background that will be used in the rest of the thesis, mainly to fix the notation and provide pointers to reference texts or papers. While starting relatively from the basics, it has no pretence to be entirely self-contained.
The following chapter deals with the topic of homology representatives. A homology representative for a class is a choice of a cycle belonging to that class, among the possibly many. This choice is notoriously ambiguous, and many different criteria can be put in place for it, but the significance of this choice can be of great importance when the applicative information lays into the actual cycles and not in their equivalence classes. Tracing back topological information onto the cycles then becomes crucial for applications. In there, we will present some applications of the algorithm by Dey concerning minimal homology bases, which are sets of cycles

which collectively span H_1 and whose total length is minimal among all possible choices. We have implemented that algorithm and used it to define a method of network skeletonization called the minimal scaffold. We will describe its properties and how the method can find valuable application in neuroscience. Then, we describe a mixed-approach for metric spaces called the homologically-persistent skeleton, study some of its properties and propose an own version to compare with the previous approach. Finally, we describe some preliminary ideas towards using of Alexander duality to find canonical representatives.

The next chapter deals with the topic of persistence module decomposition. In there, we propose the concept of an interval basis, that is a "special" choice of generators for a persistence module, such that each generates the direct summand interval module given by the structure theorem decomposition. We propose a sequence of approaches to the computation of an interval basis, and compare their scope and their cost. In particular, we propose a parallel algorithm that leads from a persistence module to an interval basis, from which it is immediate to read off the barcode. In this sense, the proposed approach is an extension of the computation of persistent homology. We subsequently specialize the discussion to the topic of persistence modules arising from the homology functor, hence setting off not from a persistence module but from a map of chain complexes. In particular, we show the construction of a parallel pipeline to compute persistent homology via the Hodge Laplacian.

A brief survey of future directions concludes the thesis.

Chapter 2

Background

2.0.1 Elements of Category Theory

We start by introducing a few basic notions in category theory, encompassing many of the later concepts in an elegant framework. Categories are *universes* that collect all objects of a certain kind, together with relations between them. One can then develop a concept of relations between relations, and relations between relations of relations, and so forth. This abstraction process leads to the concepts of functors, natural transformations and eventually higher categories. It has been postulated that natural transformations were the intuition that first sparked the birth of category theory, with functors being the object onto which they could be defined, and with categories being the *ground* object onto which to define functors. Good entry points for category theory are [1, 2] (or the joy of cats),[3] for a more advanced treatment).

Definition 1. (Category [3]) A category $\mathbf{C}$ is a collection (in general a class) of *objects*, denoted by $Ob(\mathbf{C})$, such that

- $\forall\, a,b \in Ob(\mathbf{C})$ there exists a collection (in general a class) $Hom_{\mathbf{C}}(a,b)$ whose elements are called *morphisms* or *arrows*.
- For each a, the hom-set $Hom_{\mathbf{C}}(a,a)$ contains a distinguished element called the *identity* Id_a
- There exists a *composition* function $\circ$

$$\circ \;:\; Hom_{\mathbf{C}}(b,c) \times Hom_{\mathbf{C}}(a,b) \;\longrightarrow\; Hom_{\mathbf{C}}(a,c)$$

for any triplet of objects a, b and c.

- Composition must be compatible with identities, i.e. for every $f \in Hom_{\mathbf{C}}(a,b)$ it must hold
$$f \circ \mathrm{Id}_a = f \text{ and } \mathrm{Id}_b \circ f = f$$

- Composition must be associative, i.e. for any three *composable* morphisms f, g and h it holds
$$f \circ (g \circ h) = (f \circ g) \circ h$$

An important type of categories are the so-called *small* categories: those for which objects and morphisms form a set and not a proper class. A category is at least *locally small* if, for every fixed source a and target b, $Hom_{\mathbf{C}}(a,b)$ is a set. Many important categories are *large* (that is, not small), but are locally small.

Example. The category **Set**, whose objects are sets and whose morphisms are functions between sets. The category $\mathbf{Vect}_{\mathbb{F}}$, whose objects are vector spaces over $\mathbb{F}$, and whose arrows are linear maps. These categories are not small, but they are locally small. A partial order on a set P, denoted $(P, \leq)$, is represented as a category whose objects are the elements of P, and such that there is a morphism from $p \to q$ for each relation $p \leq q$ in the poset.

The relations between categories are given in the form of *functors*, i.e. associations between the objects and morphisms of one category to those of another satisfying certain precise requirements. Functors are analogies between different universes (different categories) made precise by the present framework.

Definition 2. (Functor, [1]) Let **C** and **B** be two categories. A covariant functor $F : \mathbf{C} \to \mathbf{B}$ is a pair of functions that assign to each object of **C** an object of **B**, to each morphism in **C** a morphism in **B**, and such that identities are sent to identities and each composable pair of morphisms in **C** is sent to a composable pair in **B**, with the composition of images equal the image of the composition. In symbols

- $F \ : \ Ob(\mathbf{C}) \to Ob(\mathbf{B}) \ ; \quad a \mapsto Fa$
- $F \ : \ f \in Hom_{\mathbf{C}}(a,b) \ \mapsto Ff \in Hom_{\mathbf{B}}(Fa, Fb)$
- $F \ \mathrm{Id}_a = \mathrm{Id}_{Fa}$

- Every time that $f \circ g$ is defined for morphisms f, g in $\mathbf{C}$, then $Ff \circ Fg$ is defined in $\mathbf{B}$ and it holds

$$Ff \circ Fg = F(f \circ g)$$

We will simply say "functor" to mean a covariant functor. The notion of *contravariant* functor is the same with the exception that

$$F \ : \ f \in Hom_{\mathbf{C}}(a,b) \ \mapsto Ff \in Hom_{\mathbf{B}}(Fb, Fa)$$

and consequently the composition $Fg \circ Ff$ is defined every time $f \circ g$ exists (the functor must still commute with composition). In other words, a contravariant functor is a functor that reverses every arrow from one category to the other.

Definition 3. (Forgetful functor, [1]) The *forgetful functor*, often denoted by U, is a functor that "forgets" some or all of the additional structure on a set. As such, it associates for example to a group its underlying set and to group morphisms the corresponding functions; or to a ring the abelian group it contains.

Definition 4. (Diagram) The collection of all functors between two categories $\mathbf{C}$ and $\mathbf{B}$ is often denoted as $\mathbf{B}^{\mathbf{C}}$, and one such functor is sometimes called a *diagram of type* $\mathbf{C}$.

Example. Poset categories such as $(\mathbb{R}, \leq)$ or $([n], \leq)$ provide an important example of diagrams of type $\mathbb{R}$ or $[n]$ respectively. Diagrams of this type with value in the category $\mathbf{Vect}_{\mathbb{F}}$ of finite-dimensional $\mathbb{F}$-vector spaces correspond precisely to persistence modules.

Natural transformations are to functors what functors are to categories:

Definition 5. (Natural Transformation, [1]) Given two functors $F, G \ : \ \mathbf{C} \to \mathbf{B}$ between two fixed categories, a *natural transformation* τ is a function that associates a morphism in $\mathbf{B}$ to each object of $\mathbf{C}$, in such a way that all possible squares commute. In symbols

- $\tau \ : \ a \in Ob(\mathbf{C}) \ \mapsto \ \tau_a \in Hom_{\mathbf{B}}(Fa, Ga)$
- $\forall f \in Hom_C(a,b)$ it holds that the following diagram

$$\begin{array}{ccc} F(a) & \xrightarrow{F(f)} & F(b) \\ {\scriptstyle \tau_a}\downarrow & & \downarrow{\scriptstyle \tau_b} \\ G(a) & \xrightarrow{G(f)} & G(b) \end{array}$$

is commutative, i.e. $\tau_b\, F(f) = G(f)\, \tau_a$.

Informally, the existence of a natural transformation between functors F and G asserts the existence of all the required arrows in the target category to "transform" the action of F into the action of G while respecting the composition structure.

2.0.2 Notions of algebra

Many algebraic concepts are involved in the constructions of topological data analysis. Here, we recall some of these notions to fix definitions and notations. For module theory and linear algebra in general, we refer to [4].

Algebraically, a free object is *the most general* instance of an algebraic structure built on a given set, i.e. one containing the given set and satisfying only the axioms of that specific structure. The notion can be expressed in purely categorical terms, where a free object on set X is built via the *free functor*, that informally is the most general functor that can undo the action of the forgetful functor. First, let us introduce the notion of full anf of faithful functors.

Definition 6. (Full and faithful functors) A functor $F : \mathbf{C} \to \mathbf{D}$ is called *full* iff for every morphism g in $\mathbf{D}$ there exists a morphism f in $\mathbf{C}$ such that $g = F(f)$. It is called *faithful* iff for every pair of *parallel morphisms* f_1, f_2 in $\mathbf{C}$ (i.e. morphism between the same two objects) it holds that $F(f_1) = F(f_2)$ implies $f_1 = f_2$. When both apply, we call F a *fully faithful fuctor*. In other words, a full functor is surjective on Hom-sets, a faithful funtor is injective on Hom-sets, and a fully faithful functor is a bijection of Hom-sets.

Definition 7. (Free object) Let $\mathbf{C}$ be a concrete category, that is a category that admits a faithful functor U into **Set**, and let X be a set. A free object on X in $\mathbf{C}$ is an object A with the following universal property: for every injective map $i : X \hookrightarrow U(A)$, for every object B in $\mathbf{C}$ and for any morphism of sets $f : X \to U(B)$, there is a unique morphism of sets $g : U(A) \to U(B)$ such that f factors as $f = i \circ g$. Diagrammatically:

$$\begin{tikzcd} X \arrow[r, hook, "i"] \arrow[dr, "f"'] & U(A) \arrow[d, dashed, "g"] \\ & U(B) \end{tikzcd}$$

In words, an object A in a concrete category $\mathbf{C}$ is free if there exists a "subset" X of A such that for every object B in $\mathbf{C}$, any set function $X \to B$ extends uniquely to a morphism $A \to B$ in $Hom_{\mathbf{C}}$.

Definition 8. (Monoid) A monoid is a category with a single object.

Definition 9. (Group) A group is a monoid with inverse elements. I.e. for each morphism there is a morphism that composes to the identity on the right, and one that composes to the identity on the left.

Maps between groups must commute with the groups' operations, as in $f : G \to H$ it holds that $f(g+g') = f(g) +' f(g')$. Together with these maps, groups form the *category of groups*, **Grp**.

Definition 10. (Abelian group) We call a group Abelian if its operation is commutative.

Definition 11. (Free Group) A free group is a free object in the category **Grp** of groups.

Definition 12. (Ring) [4] A ring is a set with two operations: $(R, +, \cdot)$, such that $(R, +)$ is an Abelian group, $(R, \cdot)$ is a monoid, and $\times$ distributes over $+$ both on the left and on the right.

Other customs exist: notably, the definition above is sometimes called a *unital* ring, or ring with identity, and a more general definition of ring would only require $(R, \cdot)$ to be a semigroup, i.e. not necessarily contain an identity element. For our purposes, every ring will be unital, so we include the requirement of an identity element into the definition of ring.

Example. The integers $(\mathbb{Z}, +, \cdot)$, with sum and multiplication, are the prototypical example of a ring (in fact, they are the *initial object* in the category **Ring** of unital rings). Another typical example is the ring of polynomials in n variables with integers coefficients $\mathbb{Z}[x_1, \ldots, x_n]$, with pointwise sum and multiplication.

Definition 13. (Initial, terminal and zero object) [1] An object $s \in Ob(\mathbf{C})$ is called *initial* iff for every object $c \in Ob(\mathbf{C})$ there exists exactly one morphism $s \to c$. $t \in Ob(\mathbf{C})$ is called *terminal* iff for every object $c \in Ob(\mathbf{C})$ there exists exactly one morphism $c \to t$. Initial and terminal object are unique up to unique isomorphism. An object that is both initial and terminal is called a *zero object*.

It is customary to call a *commutative ring* a ring such that multiplication is also commutative. Again, in all of the following we will only deal with commutative rings, so unless stated otherwise, when we write ring we actually mean (in full) a *commutative ring with identity*. Notice both the examples above are also examples of commutative rings.
Unitary, although not necessarily commutative rings whose non-zero elements form a monoid under multiplication are called *integral domains*, the key being that multiplications of non-zero elements yields non-zero elements. In other words, there are no zero-divisors.
If additionally the non-zero elements form a group under multiplication, the ring is called a *division ring*. A division ring is a non-commutative field. An element of a ring that admits an inverse is called a *unit*. So a division ring is a ring such that all non-zero elements are units.

Definition 14. (Field) A field $\mathbb{F}$ is a commutative ring such that multiplication admits an inverse for every element, except for the additive identity. In other words, it is a commutative division ring.

Example. The reals, the rationals, the complex numbers, and notably the integers modulo p with p a prime number are examples of fields. $\mathbb{Z}_2$ in particular plays a central role in most of TDA.

Finally, let us introduce a special type of ring that will play a key role in the following. By *ideal* I of a ring R we mean an additive subgroup that is stable under multiplications by elements of R, i.e. such that $ir \in I$ for every $i \in I, r \in R$.

Definition 15. (PID, [4]) We call Principal Ideal Domain (PID) a commutative unital ring with no zero divisors (i.e. a commutative integral domain), such that every ideal is *principal*, i.e. such that every ideal is of the form aR for some $a \in R$.

Principal ideals generalize the notion of ideals in the ring of integers, indeed a PID, where every product-stable subgroup is the set of multiples of some integer number.

Example. Polynomial rings over a field In the following, we will often consider the ring of polynomials in n variables with coefficients in a field, $\mathbb{F}[x_1,\dots,x_n]$. This is a unital, commutative ring. The case $n = 1$ is especially important, because $\mathbb{F}[x]$ is a principal ideal domain.

Definition 16. (Module) [4] Let R be a ring (unital and commutative), with 1 its multiplicative identity. A module M over R (or an R-module) is an Abelian group $(M,+)$, together with an action $R \times M \to M$, called scalar multiplication, with the following properties: $\forall \lambda, \mu \in R$, $\forall x, y \in M$

- $\lambda(x+y) = \lambda x + \lambda y$
- $(\lambda + \mu)x = \lambda x + \mu x$
- $\lambda(\mu x) = (\lambda\mu)x$
- $1x = x$

Since we only deal with commutative rings, we do not define left and right modules. Note that a ring R is an R-module over itself.

Definition 17. (Vector Space) Given a field $\mathbb{F}$, a vector space V over $\mathbb{F}$ (or an $\mathbb{F}$-vector space) is an $\mathbb{F}$-module.

So a vector space is a module over a field instead of a ring.
Modules and vector spaces are connected by *linear maps*. A linear map is a map that commutes with *linear combinations*, i.e. one such that $f(\lambda_1 v_1 + \cdots + \lambda_n v_n) = \lambda_1 f(v_1) + \cdots + \lambda_n f(v_n)$ where the λ's are elements of a ring or a field, respectively. Together with linear maps, R-modules form the category **RMod**, and $\mathbb{F}$-vector spaces form the category $\mathbf{Vect}_{\mathbb{F}}$.
The concept of free object also applies to R-modules.

Definition 18. (Free Module) A free R-module is a free object in the category **RMod**.

Alternatively, for a free module F on set G, there exists a unique map h such that the diagram

$$\begin{tikzcd} G \arrow[r, "f"] \arrow[dr, "g"'] & F \arrow[d, dashed, "h"] \\ & M \end{tikzcd}$$

is commutative for any R-module M and any mapping g. Intuitively, if a module is free then maps from its underlying set can be used to fully determine maps from the module itself.
If a module F is free (or more precisely the pair (F, f) is a free module), then f is injective. Furthermore, $\operatorname{Im} f$ is a *basis* of F, i.e. a linearly independent set that generates it.
In fact, for an R-module possessing a basis is equivalent to being free. As such, every vector space is a free module.

Given a subset G of a module M, we say that G generates M if the linear span of G, i.e. the module of the linear combinations of elements in G, equals M. The elements of G are called generators.

Definition 19. (Finitely generated module) We say a module is *finitely generated* if it admits a finite set of generators.

Grading The concept of grading denotes the existence of a direct sum decomposition of an object into "graded" subparts. Reference also [5–7].

Definition 20. (Graded Ring) An $\mathbb{N}$-graded ring is a ring R for which one has a direct sum decomposition

$$R = \bigoplus_{i \in \mathbb{N}} R_i$$

as Abelian groups, such that $R_i R_j \subset R_{i+j}$, that is, the product of an element in R_i with an element in R_j is an element of R_{i+j}.

An element of R belonging to some R_i is called *homogeneous*. Notice 0 is a homogeneous element. For a non-zero homogeneous element $r \in R$, there is only one i such that $r \in R_i$, and we say r is *of degree i*. The degree of 0 is not defined.
A morphism of graded rings is a ring homomorphism $\varphi : R \to R'$ such that $\varphi(R_i) \subseteq R'_i$.

Definition 21. (Graded Module) Let R be an $\mathbb{N}$-graded, unital, commutative ring, and let M be an R-module. M is graded if one has

$$M = \bigoplus_{i \in \mathbb{N}} M_i$$

as Abelian groups, and it holds $R_i M_j \subseteq M_{i+j}$

Notice that every module can be regarded as a (trivially-) graded module by setting above $M_1 = M$ and $M_i = \{0\}\ \forall i > 1$. Notice also that a graded module can be given different gradings.
A morphism of graded modules is an R-linear map $\varphi : M \to N$ such that $\varphi(M_i) \subseteq N_i$. It is also referred to as *graded morphism* or *graded map*.
A basis of a graded module is required to be homogeneous, i.e. that each basis element is homogeneous.
Graded modules together with graded maps form the *category of graded modules* **GRMod**.

Shifting

Graded maps between graded modules are defined to be *of degree zero*, i.e. such that grading is preserved when the map is applied. Therefore ([7]), two free, rank-1, graded modules need not be isomorphic via a graded isomorphism: this could only be the case if their generator was the same degree for both. One could define a map *of degree n*, that is a graded map that sends elements of degree i to elements of degree $i+n$. Or, as is standard ([8, 5]), one could define a *shifted* module in the following way: let R be any ring,

$$R[-n]$$

is the free, graded module over one homogeneous generator of degree n, i.e. such that homogeneous elements of degree n in $R[-n]$ are the elements of degree 0 in R (seen as a rank-1, free module over itself).

Presentations

As we said, free modules allow for a basis, i.e. a way to specify a map from the free module F as a map from a set B. In the following, consider a *finitely generated* module M *over a principal ideal domain*. Then M can be specified via a surjective map μ from a free module G, called the *module of generators*

$$G \xrightarrow{\mu} M$$

Intuitively, map μ specifies how the generators of G map onto the target module, that is what the linear dependencies are between the generators in the target object. In particular, the kernel $\ker\mu$ contains (if any) the *relations* that fully characterize module M. The presence of relations between the generators indicates the presence of *torsion* in M

$\ker\mu$ is a subobject (a submodule) of G, and in particular it fits into a short exact sequence of the form

$$0 \longrightarrow \ker\mu \xhookrightarrow{i} G \xrightarrow{\mu} M \longrightarrow 0 \tag{2.1}$$

Notice that, for any module M, many different such sequences may exist.

A *complex* of R-modules ([9]) is a sequence of R-modules M_i and maps $M_i \to M_{i-1}$ such that $M_i \to M_{i-1} \to M_{i-2}$ is the zero map. A *free resolution* of an R-module M is a complex of *free* R-modules that is exact, and such that $M = \text{coker } (M_1 \to M_0)$. We say a free resolution has length n if the M_i are non-trivial for $0 \leq i \leq n$ and M_n is zero.

Hilbert's Syzygy Theorem ensures us that, since the ground ring is a PID, a free resolution of M has length 1, which means that, in the sequence above, map i is injective. This is the case for polynomial rings in one variable over a field ($\mathbb{F}[x]$) that encompass the theory of one-dimensional persistence.

Notice that, for polynomial rings in several variables, that are their counterpart in multi-dimensional persistence and which do not form principal ideal domains, this no longer holds: one could have that the map $\ker\mu \to G$ is not injective, hence has a nontrivial kernel giving rise to further relations between relations, and possibly so on.

Definition 22. (Presentation) Giving a *presentation* of M amounts to specifying a short exact sequence as in 2.1

As mentioned above, $\ker\mu$ specifies how the generators of M are related, or in other words describe the torsion part in M. This way, a module can be expressed as the coker of map i:

$$M \cong \frac{G}{\text{Im } i} = \text{coker } i \tag{2.2}$$

Definition 23. (Relations or Syzygies) The generators of $\ker\mu$ (as a free submodule of G) are called *syzygies*.

In the case when the ground ring is not a PID, i.e. when a free resolution of M may have length > 1, we call the relations between relations *second syzygies*, and so on with third syzygies, etcetera.

Definition 24. (Minimal Presentation) A presentation is called *minimal* if module G (and hence $\ker \mu$) is the smallest possible, in terms of the number of generators.

Notice that for a free module over a PID, all bases have the same cardinality.
All of the above can be cast in the language of graded modules, by requiring that all of M, G and $\ker \mu$ be graded, finitely generated modules over a PID, and that all maps be graded morphisms. This is the case for the modules arising from persistent homology.

Finding a module presentation

Finitely generated modules over principal ideal domains are particularly well-behaved. Since the ground ring is especially tame, as is the case for $\mathbb{F}[x]$, the structure of their modules can be described in an explicit manner, thanks to the following theorem.

Theorem 1. *Structure Theorem for modules over* $\mathbb{F}[x]$, *[4]*

$$M \cong \bigoplus_{i=1}^{n_F} \mathbb{F}[x] \oplus \bigoplus_{j=1}^{n_T} \mathbb{F}[x]/(\alpha_j) \tag{2.3}$$

The above theorem states that a module can be decomposed into the direct sum of n_F copies of the free, rank-1 module $\mathbb{F}[x]$ (again seeing the ring as a module over itself), plus a sequence of n_T cyclic modules of order α_j. This is the torsion part.
Further, this decomposition cannot be refined, as each term in the direct sum is easily seen to be *indecomposable* (we call an R-module indecomposable if it cannot be expressed as a direct sum of two non-zero submodules).
This decomposition is intimately related to a matrix canonical form known as *Smith Normal Form*.

Definition 25. (Smith Normal Form) Let R be a PID. Let A be an $n \times m$ matrix with entries in R. Then, there exist two square matrices $M \in R^{m \times m}$ and $N \in R^{n \times n}$ such that NAM is diagonal, and its non-zero diagonal entries are such that $\alpha_i | \alpha_{i+1}$. Finally, all zero entries appear at the end.

The diagonal entries are unique up to a unit, an invertible element of the ring.
The Smith normal form is a central result in linear algebra, and it condenses all the information we can gather about a module over a PID, up to isomorphism.

A corresponding theory applies to the case of *graded modules* over $\mathbb{F}[x]$, which is graded by polynomial degree

Theorem 2. *Structure Theorem for Graded modules over* $\mathbb{F}[x]$

$$M \cong \bigoplus_{i=1}^{n_f} x^{\alpha_i}\, \mathbb{F}[x][\alpha_i] \oplus \bigoplus_{j=1}^{n_t} x^{\beta_j}\, \mathbb{F}[x][\beta_j]/(\beta_j + \gamma_j) \tag{2.4}$$

and this instance is of central importance for topological data analysis, as we will see in the following. A graded theory of the Smith Normal Form has also been advanced, see for example [10]. This framework will prove relevant for our purposes, as the role of grading is central for persistence modules.

2.0.3 Simplicial complexes

The most fundamental building block around which applied topology is constructed is, arguably, the concept of simplicial complex. Simplicial complexes are *discrete* topological, and possibly geometrical, objects, which encode a shape much in the same way that an adjacency matrix represents a graph, i.e. by specifying how blocks of a certain dimension are attached to blocks of the next.
From the most basic of points of view, one can think of a simplicial complex as a discretisation of a manifold; this point of view highlights the fact that, as discrete objects, they allow a manifold to be approximately represented on a computer machine. One could alternatively view a simplicial complex as a decorated graph, i.e. a graph to which one adds higher-dimensional features by the same attaching mechanism that is commonly used for edges and vertices.
Simplicial complexes come in essentially two flavours, foreshadowed above: they can be endowed with geometric information, by being embedded in some $\mathbb{R}^n$ and built from a set of its points; this flavour mostly reminds the reader of the manifold metaphor. Or they can be regarded as purely combinatorial objects, that do not live in any particular space and only encode adjacency information; this is instead the

graph-theoretic flavour.
We shall begin to describe them from the geometric point of view, and later see how the abstract case generalises it.

Definition 26. (Geometric independence) $k+1$ points $v_0, \ldots, v_k$ in the affine space $\mathbb{R}^n$ are said *geometrically independent* if the vectors $v_0\bar{v}_i$ are linearly independent.

Notice the definition does not depend upon the ordering of the points.
The *convex hull* of points $v_0, \ldots, v_k$ is the set of their *convex combinations*, i.e. of sums of the form

$$\sum_{i=0}^{k} \lambda_i v_i \quad \text{where} \quad \lambda_i \geq 0 \quad \text{and} \quad \sum_{i=0}^{k} \lambda_i = 1$$

Convex combinations generalize the concept of weighted averages.

Definition 27. (Simplex) A *k-simplex* is the convex hull of $k+1$ geometrically independent points.

For a k-simplex σ we write $\dim \sigma = k$ and say that the *dimension* of σ is k.
As a consequence of the property of geometric independence of points $v_0, \ldots, v_k$, the parameters λ_i's form a set of *coordinates* for the points belonging to a simplex, called *barycentric coordinates*.
The convex hull of a single point is the point itself, and that is a 0-simplex. For two geometrically independent (i.e. distinct) points, their convex hull is a segment, which settles the case for 1-simplices. Geometric independence of three points amounts to non-collinearity, therefore a 2-simplex is a triangle. Finally, four points not lying on the same plane are geometrically independent, therefore 3-simplices are tetrahedra. Higher-dimensional counterparts eventually fail our intuition.
So a geometric simplex σ is the convex hull of a set of points. The convex hull of a (non empty) subset of the points of σ is another simplex, which we call a *face* of σ. We write $\tau \preceq \sigma$ to mean that τ is a face of σ, and conversely σ is a *coface* of τ ($\sigma \succeq \tau$). Notice a face is guaranteed to be a simplex because a subset of geometrically independent points is again geometrically independent.
Simplices are arranged together to form the landscape of applied topology, but that arrangement requires some assumptions

Definition 28. (Geometric simplicial complex) A (geometric) *simplicial complex K* is a set of (geometric) simplices such that

1. If $\sigma \in K$ and $\tau \preceq \sigma$, then $\tau \in \sigma$.
2. If $\sigma, \tau \in K$ then their intersection is either empty or a face of both.

The first property requires that no building block is left hanging, i.e. that if a block of a certain dimension exists in the complex then all of its lower dimensional sub-blocks are also present. The second property requires proper gluing between blocks, i.e. that no two elements in the complex intersect along missing sub-objects.

Geometric complex are generalized by the notion of abstract simplicial complexes, whereby we drop the assumptions about the basic building blocks being points in a space $\mathbb{R}^n$ and instead assume the existence of a base set of elements that we call *vertices*.

Definition 29. (Abstract Simplicial Complex) Let V be a finite set, whose elements we call vertices. Consider its power set 2^V, as a poset ordered by subset inclusion $\subseteq$. An abstract simplicial complex on V is a downward-closed family of subsets of 2^V.

We see that in the definition of abstract simplicial complex we retain the requirement that the complex be closed under restriction. On the other hand, we do not need to enforce that the geometric intersections and the set-theoretic intersections agree, as we have dropped the geometric information altogether.
The two notions are somewhat dual: given a geometric complex, it suffices to add to V an element $\tilde{v}$ for each vertex v, and an abstract simplex $\{\tilde{v}_0 \dots \tilde{v}_k\}$ for each geometric simplex that is the convex hull of points $v_0 \dots v_k$. That is, an abstract simplicial complex simply *forgets* the geometric information.
Dually, given an abstract complex K, it suffices to embed each element of V into a point in space, so that the image of this map is geometrically independent. Then,

Definition 30. (Geometric Realization) The geometric complex obtained by considering the convex hulls of the image of each element of K is called a *geometric realization* of K, denoted by $|K|$.

Definition 31. (Subcomplex) Given an abstract simplicial complex K, a subset $K' \subseteq K$ which is itself a simplicial complex is called a *subcomplex* of K. We write $K' \trianglelefteq K$.

Definition 32. (k-skeleton) We call the k-skeleton of a simplicial complex the set of its k-simplices.

2.0.4 Homology

Homology is a topological tool which provides invariants for shape description and characterization. It relies on associating an algebraic object to the topological or possibly geometrical objects that we have seen above. This allows to recast questions about geometry as questions involving linear algebra, a subject that is amenable to being handled by a computing machine.

Let K be a finite abstract simplicial complex. The first step requires that we fix an *orientation* on its simplices. We recall a simplex is a set $\sigma = \{v_0, \dots, v_k\}$

Definition 33. (Orientation) An *orientation* on a simplex is an equivalence class of permutations of its vertices up to parity, denoted by $[v_0 \dots v_k]$.

So, for any simplex there exist only two possible orientations. An *oriented simplex* is a simplex together with a specified ordering of its vertices. We will write an oriented simplex as $[v_0 \dots v_k]$, so that one can speak of positive and negative orientations, according to the parity of the permutation with respect to the given one. Given a simplicial complex, it is possible to fix an orientation that is compatible with the face relations. A standard way to give such an orientation is to give a total order on the vertices, and orient each simplex monotonically.
The existence of a positive and negative orientation on each simplex allows for the definition of an Abelian group of *oriented k-simplices*, where the only relations are given by $\sigma + \tau = 0$ if and only if σ and τ are the same simplex with opposite orientation.

Definition 34. (k-chains) Let R be a (commutative, unital) ring and K an oriented simplicial complex. We call the module of k-chains of K the free R-module over its oriented k-skeleton, where the Abelian group structure is given as above. We denote it by $C_k(K,R)$.

We will normally fix as the standard basis of the free module the one corresponding to each oriented k-simplex.
Throughout the rest of this work, we will have the base ring actually be a field $\mathbb{F}$, normally the finite field of integers modulo 2. Therefore, k-chain will in fact form an $\mathbb{F}$-vector space $C_k(K,\mathbb{F})$, also denoted by $C_k(K)$ when the base field is clear from context.

The boundary operator connects each $C_k(K)$ to $C_{k-1}(K)$

Definition 35. (Boundary operator) We call *boundary operator* the linear map of free modules

$$\partial_k \;:\; C_k(K) \longrightarrow C_{k-1}(K)$$

that acts on each basis element as

$$\partial_k[v_0 \dots v_k] \;=\; \sum_{i=0}^{k} (-1)^i \, [v_0 \dots \hat{v}_i \dots v_k]$$

where $[v_0 \dots \hat{v}_i \dots v_k]$ denotes removing vertex v_i.

Definition 36. (Chain complex) Given a finite, abstract, oriented simplicial complex K, a chain complex is the sequence of free R-modules and morphisms

$$\dots \xrightarrow{\partial_{k+1}} C_k(K) \xrightarrow{\partial_k} C_{k-1}(K) \xrightarrow{\partial_{k-1}} \dots \xrightarrow{\partial_1} C_0(K) \xrightarrow{\partial_0} 0$$

where each chain group is mapped to the next one through the boundary operator.

Again, for the remainder of our discussion all of these modules will in fact form $\mathbb{F}$-vector spaces. From now on, this assumption will be valid throughout.

Definition 37. (Cycles) The space of k-cycles is the subspace of $C_k(K)$ given by $\ker \partial_k$, and is denoted by $Z_k(K)$.

Definition 38. (Boundaries) The space of k-cycles is the subspace of $C_k(K)$ given by $\mathrm{Im}\ \partial_{k+1}$, and is denoted by $B_k(K)$.

Lemma 1. *It holds that $\partial_k \partial_{k+1} = 0$.*

Proof. It suffices to repeatedly apply the definition. $\square$

Therefore, Im $\partial_{k+1} \subseteq \ker \partial_k$ as vector subspaces. It therefore makes sense to consider their quotient.

Definition 39. (Homology) We call k-homology group of K over a field $\mathbb{F}$ the quotient vector space of $Z_k(K)$ over $B_k(K)$.

$$H_k(K,\mathbb{F}) \;=\; \ker \partial_k \;/\; \mathrm{Im}\; \partial_{k+1}$$

We will call two k-cycles *homologous* if they belong to the same homology class. Roughly speaking, homology reveals the presence of “holes" in a shape. A non-null element of $H_k(K)$ is an equivalence class of cycles that are not the boundary of any collection of $(k+1)$-simplices of K. Such classes represent, in dimension 0, the connected components of complex K, in dimension 1, the holes punched in its surface, in dimension 2, the voids or cavities, and so on.

Abstract simplicial complexes form a category **Simp**, whose morphisms are simplicial maps. Simplicial maps (maps between simplicial complexes K and K') are induced by a vertex map, i.e. a map of the 0-skeleton of K such that if a set of vertices is a simplex in K, its image is a simplex in K'. These maps induce in the obvious way a graded homomorphism between the chain complexes of K and K'. For every $k \geq 0$, homology in degree k over a field $\mathbb{F}$ assigns to a simplicial complex K an $\mathbb{F}$-vector space $H_k(K,\mathbb{F})$, i.e. an element of the category $\mathbf{Vect}_{\mathbb{F}}$. This assignment is in fact a functor: homology in degree k is a functor $H_k \;:\; \mathbf{Simp} \to \mathbf{Vect}_{\mathbb{F}}$, which means that a simplicial map $f \;:\; K \to K'$ induces uniquely a linear map of $\mathbb{F}$-vector spaces between homology groups, denoted $f^* \;:\; H_k(K,\mathbb{F}) \to H_k(K',\mathbb{F})$, in such a way that all squares commute.

2.0.5 Complexes from Data

Applying topological methods into data analysis requires methods to build simplicial complexes from data. Here, we overview a few of the most employed schemes.
A *metric space* (respectively, *extended* metric space) is a pair (X,d), where X is a set and d maps pairs of elements of X into $\mathbb{R}_{\geq 0}$ (respectively $\mathbb{R}_{\geq 0} \cup \{+\infty\}$), with the known properties: that d be symmetric, and zero only over all of the diagonal, and satisfying the triangle inequality.

A metric that satisfies all of the above, except possibly for the triangle inequality is called a *semi-metric*. A *pseudo-metric* is a metric that is zero over all of, but not necessarily only on, the diagonal of $X \times X$.
In the following, we shall normally deal with data that comes in one of two forms: as a discrete subset of a normed vector space $\mathbb{R}^n$, therefore as a metric subspace, called a *point cloud*; or as a non-negatively weighted, undirected, self-loop free graph, which forms a (pseudo-, semi-) metric space in itself. Crucially, one is eventually presented with a matrix $d_{i,j}$, encoding a *dissimilarity* measure between entities i and j.
Once in possession of either a point cloud or a dissimilarity graph, several methods exist to turn this information into topology. The *nerve* of a cover is the most classical tool to turn a covering into a simplicial complex.

Definition 40. (Nerve Complex) Let S be a family of subsets of $\mathbb{R}^n$. We call the *nerve of S* the simplicial complex N built by introducing a k-simplex for each non-empty k-fold intersection between elements of S.

It is immediate to see that N is indeed a simplicial complex.
Under suitable conditions on the family of subsets S, some version of the *nerve lemma* applies, guaranteeing that the topology of the nerve is "close" to the topology of the union of S.
Now assume a point cloud Q is given in $\mathbb{R}^n$, let $\varepsilon \geq 0$, and consider as a family S_ε the set of closed balls of radius ε centered at each point of Q.

Definition 41. (Čech Complex) We call the *Čech complex* of point cloud Q at scale ε the nerve of the above cover S_ε, denoted by $\check{C}_\varepsilon(Q)$.

Due to the symmetric nature of balls in $\mathbb{R}^n$, the Čech complex can equally be built just by knowing the distance matrix $d_{i,j}$.
Given the good properties of closed balls in $\mathbb{R}^n$, the nerve lemma applies for the Čech complex; however, in practical applications it is often too expensive to compute the nerve of a large point cloud, as checking k-fold intersections requires a number of distance evaluations that grows as the k^{th} power of the number of points.
Another option, which instead requires the knowledge of an embedding of the point cloud Q into its ambient space $\mathbb{R}^n$, is the *Alpha complex*.

Definition 42. (Voronoi cell) [11] Given a point cloud Q in $\mathbb{R}^n$ and $x \in Q$, the *Voronoi cell* of x is the set of points of $\mathbb{R}^n$ for which x is the closest among all points of Q.

$$V(x) := \{y \in \mathbb{R}^n \;/\; \|y-x\| \leq \|y-z\|,\, z \in Q\}$$

The collections of the Voronoi cells of the points in Q is the Voronoi *diagram* of Q.

Let $V(p)$ be the Voronoi region associated to point p in Q, and consider the closed ε-balls around p as above. Let $A := \bigcup_{p \in Q} B_\varepsilon(p) \cap V(p)$.

Definition 43. (Alpha Complex) The *alpha complex* of point cloud Q at scale ε is the nerve of cover A, denoted $A_\varepsilon(Q)$.

Another option to simplify the association of a complex to data is the *flag complex*. The issue of checking all k-fold intersections is solved by stopping at $k = 2$, and then including *all* simplices that are compatible with that edge structure. Consider any graph G over a set of vertices as the 0- and 1-skeleton of a simplicial complex

Definition 44. (Flag Complex) The flag complex of G is *the largest* simplicial complex that is compatible with that 1-skeleton.

Now let $\varepsilon \geq 0$, consider as above a point cloud Q, and the family of closed balls $S := \bigcup_{p \in Q} B_\varepsilon(p)$. Consider the graph obtained by applying the nerve construction to S, *up to dimension* 1. In other words, let G be the graph over the set Q considered as vertices, and containing an edge (i, j) if and only if the distance between points i and j does not exceed 2ε.

Definition 45. (Vietoris-Rips Complex) The Vietoris-Rips complex of Q at scale ε is the flag complex of the above graph G.

Notice that all of the above constructions depend on the choice of a value for parameter ε. This property will be crucial in the next section.

2.0.6 Persistent homology

Definition 46. (Filtration) A *filtration* of simplicial complex is a functor $\mathcal{F}$ from a poset category into the category of simplicial complexes, that assigns to every arrow $u \leq v$ in the poset the inclusion $\mathcal{F}(u) \subseteq \mathcal{F}(v)$.

By functoriality, the inclusion between step u and v factors through any intermediate step.
The filtrations we will consider map from a totally ordered set, typically either the real numbers $(\mathbb{R}, \leq)$ or a finite, discrete, linearly ordered set such as a subset of the naturals $([n], \leq)$.
One can equivalently consider the inclusions $\mathcal{F}(u) \subseteq \mathcal{F}(v)$ as injective maps $i : \mathcal{F}(u) \hookrightarrow \mathcal{F}(v)$. Calling a generic poset $(P, \leq)$, a filtration is a *diagram* of type $(P, \leq)$ in the category of simplicial complexes, denoted as $\mathbf{Simp}^{(P,\leq)}$.

Several ways exist to construct filtrations of simplicial complexes.

Definition 47. (Filtering Function) Let K be a simplicial complex. A *filtering function* f on K is a real-valued function such that if $\tau \preceq \sigma$ in K, then $f(\tau) \leq f(\sigma)$.

Example 1. (Filtration from a filtering function) Let K be a simplicial complex, and f a filtering function on K. For each $r \in \mathbb{R}$, the preimage $f^{-1}((-\infty, r])$ is a simplicial complex by the definition of filtering functions, and further if $r \leq s$ then $f^{-1}((-\infty, r]) \subseteq f^{-1}((-\infty, s])$. This is a functor from the poset of inclusions $(-\infty, r], \subseteq (-\infty, s]$ into simplicial complexes, mapping inclusion of intervals into the inclusion $K^r \subseteq K^s$ (where we denote by K^r the sublevel set $f^{-1}((-\infty, r])$). It is a functor $\mathcal{F}$ from the poset $(\mathbb{R}, \leq)$ into simplicial complexes, and specifically a filtration.

Definition 48. (Sublevel set filtration) A filtration built as above is called a *sublevel set filtration*.

Example 2. (Čech and Vietoris-Rips filtrations) Let Q be a point cloud, and for each $\varepsilon \geq 0$ consider the Čech complex $\check{C}_\varepsilon(Q)$ and the Vietoris-Rips complex $VR_\varepsilon(Q)$. For each arrow in the poset of the reals $\varepsilon \leq \varepsilon'$ consider the inclusions $\check{C}_\varepsilon(Q) \subseteq \check{C}_{\varepsilon'}(Q)$ and $VR_\varepsilon(Q) \subseteq VR_{\varepsilon'}(Q)$. These assignments constitute a functor from $(\mathbb{R}, \leq)$ into **Simp**, i.e. filtrations, and are called respectively *Čech* and *Vietoris-Rips filtrations*.

So a filtration is a (type of) functor from a poset into simplicial complexes, and homology can be construed as a functor from simplicial complexes into $\mathbb{F}$-vector spaces (Bubenik Categorization of PH). Combining the two yields the notion of persistent homology.

Definition 49. (Persistence module) Let $\mathcal{F}$ be a filtration of simplicial complexes. We call k^{th} persistence module the composition of $\mathcal{F}$ with the k^{th} homology functor

H_k

$$PH_k \ := \ H_k \circ \mathcal{F}$$

A persistence module is therefore a diagram of vector spaces indexed by a poset, such as $\mathbf{Vect}_{\mathbb{F}}^{(\mathbb{R},\leq)}$ or $\mathbf{Vect}_{\mathbb{F}}^{([n],\leq)}$.
In particular, the k-homology functor induces a map $i^*_{u,v}$ for each inclusion $i_{u,v}$: $\mathcal{F}(u) \hookrightarrow \mathcal{F}(v)$. These induced maps are what is used to define persistent homology groups.

Definition 50. (Persistent homology groups) We call k^{th} *persistent homology group* from step u to step v the vector space

$$PH_k(u,v) \ := \ \text{Im } i^*_{u,v}$$

Space $PH_k(u,v)$ is generated by those homological features that existed at step u, and are not mapped to zero along the path into v. We say that *are still alive at v*.

An $\mathbb{N}$-indexed persistence module is called *tame* if only a finite number of induced maps $i^*_{u,v} = H_k\mathcal{F}(u \leq v)$ are not isomorphisms, and each $H_k\mathcal{F}(u)$ is finite-dimensional. For an $\mathbb{R}$-indexed persistence module, we say a value u is *regular* if there exists an open interval containing u where the map $i^*_{x,y}$ is an isomorphism for each x, y in the interval; u is called *critical* otherwise. An $\mathbb{R}$-valued persistence module is tame if each step is finite-dimensional and the set of critial values is finite. Notice that if a real-indexed persistence module is tame, then one can regard it essentially as if it were indexed on a discrete, finite set $[n]$. The same applies to $\mathbb{N}$-indexed modules. We will deal in practice only with this type of objects.

The name persistence module is also employed to describe different incarnations of the same concept. As well as a functor from posets into vector spaces, one can equally regard a tame persistence module as a family of vector spaces connected by morphisms, as in [12].

Definition 51. (DAPM, [12]) A *discrete algebraic persistence module* (DAPM for short) is a pair $\mathcal{M} = (M_i, \varphi_{i,j})$ for $i, j \in \mathbb{N}$ and $i \leq j$, where $\varphi_{i,j} \ : M_i \to M_j$, $\varphi_{i,i}$ is the identity and for any three steps i, j, h the maps factor functorially.

The two concepts of persistence module are indeed equivalent ([13]).
The fundamental theorem of persistent homology, due to Zomorodian and Carlsson, is what allows to employ tools in representation theory to make computations about persistence modules.

Theorem 3. *(Zomorodian-Carlsson, [8]) The category of tame DAPM over a field $\mathbb{F}$ is isomorphic to the category of $\mathbb{F}[x]$-modules.*

The correspondence is given as a functor α that associates to the DAPM $\mathcal{M}$ a graded $\mathbb{F}[x]$-module $\alpha(\mathcal{M})$, obtained by considering the direct sum of homogeneous components $\bigoplus_i M_i$, and setting the action of x on any homogeneous, degree-i, element m_i as $x\, m_i = \varphi_{i,i+1}(m_i)$. In words, multiplication by x sends each element forward one step along the graded maps.

Corbet and Kerber ([12]) generalize this result to persistence modules over a ring R, with the additional requirement that the persistence module be *finitely presented*.

This result is of primary importance, because it allows to leverage the theory of graded module, and specifically the structure theorem (2.4). Implied by this result is the fact that every persistence module is completely described up to isomorphism, by a set of numbers. These pairs are either of the form $(\alpha_j, +\infty)$, which are called *essential pairs* and correspond to the free generators in the decomposition theorem, i.e. to zero elements in the Smith Normal Form of a module presentation, or of the form $(\beta_i, \beta_i + \gamma_i)$ (more frequently denoted by (b_i, d_i)), which are instead called *regular pairs* and correspond to torsion elements in the module. Their corresponding entry in the Smith Normal Form is non-zero, and we distinguish between the cases where it is a non-invertible element, which gives rise to a non-degenerate pair, and the case where it is a unit, giving rise to a zero-persistence pair (b_i, b_i). The latter correspond to surplus generators, and are normally discarded.

Definition 52. (Persistence diagram) We call *persistence diagram* the set of persistence pairs as above.

It is a multiset of points in $\mathbb{R}^2$, lying above the main diagonal. An equivalent representation is called the *barcode*, consisting of a set of horizontal bars, one for each pair, spanning from b_i to d_i on the x-axis.

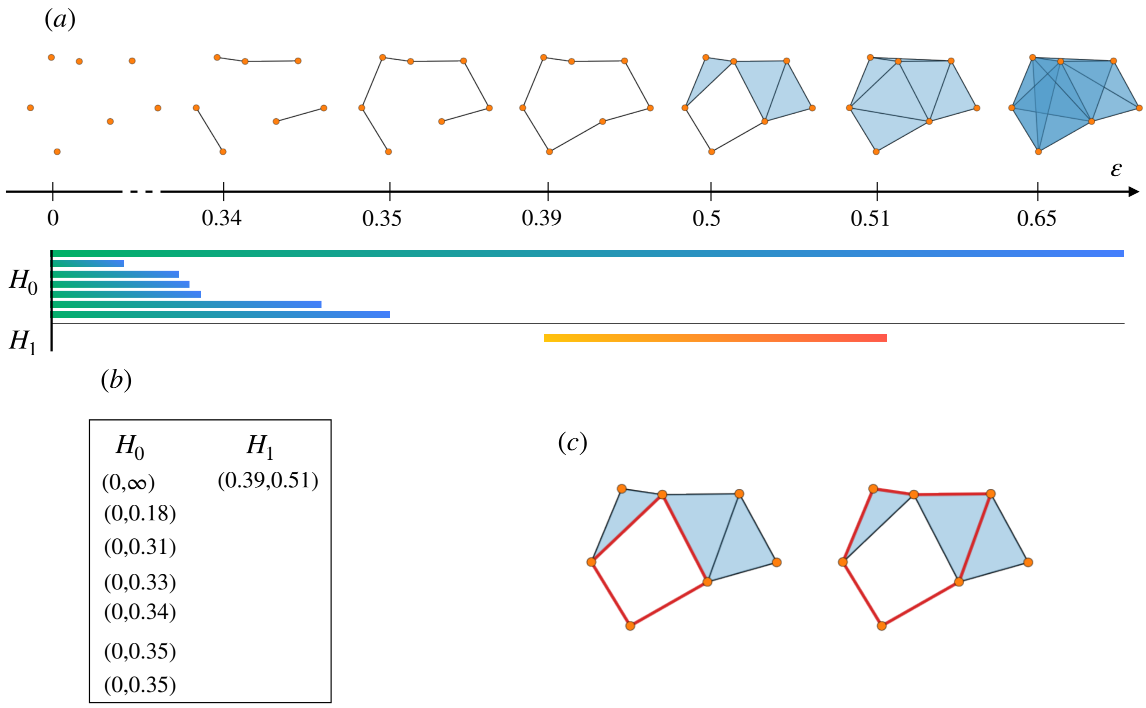


Fig. 2.1 (a) An example of Vietoris-Rips filtration of simplicial complexes with parameter ε, and the corresponding barcode for 0- and 1-dimensional persistent homology. (b) The persistence pairs of the above filtration. (c) Two equivalent representatives of the (only) generator of PH_1.

Notice all this works k by k, but it is commonplace to consider persistence diagrams or barcodes containing pairs coming from homology in several dimensions.

2.0.7 Interval Modules and the Krull-Schmidt decomposition

Definition 53. An *interval module* is a diagram $\mathcal{I}_I$ in $\mathbf{Vect}_{\mathbb{F}}^{(\mathbb{R},\leq)}$ of the following form: let I be an interval in $\mathbb{R}$; to each element $x \in \mathbb{R}$ the functor $\mathcal{I}_I$ associates

$$\mathcal{I}_I(x) = \begin{cases} \mathbb{F} & \text{if } x \in I \\ 0 & \text{otherwise} \end{cases}$$

and to each morphism $x \leq y$

$$\mathcal{I}_I(x \leq y) = \begin{cases} \mathrm{Id}_{\mathbb{F}} & \text{if } x, y \in I \\ 0 & \text{otherwise} \end{cases}$$

A similar definition can be given for integer intervals, where we substitute for I the set $[b,d] = \{b, b+1, \ldots, d\}$, possibly with $d = +\infty$.

A module M is called *indecomposable* if no two nonzero submodules exist such that their direct sum is M. A submodule N of M is called a *summand* if there exists a nonzero submodule N' of M such that $M = N \oplus N'$.
Interval modules are indecomposable as persistence modules. ([14]). The structure theorem for persistence modules entails that any persistence module can be decomposed into a direct sum of indecomposables, and that these indecomposables are exactly the interval modules as above. Furthermore, by the Krull-Remak-Schmidt theorem, this decomposition is *unique* up to isomorphism and up to a reordering of the terms.
A persistence module is called of *finite type* if it allows for a direct sum decomposition into interval modules. For persistence modules over a totally ordered set, this is equivalent to a persistence module being tame.

Chapter 3

Canonicity of Homology Generators

This chapter is based on work that I carried out together with my supervisor Francesco Vaccarino, in collaboration with Alessandro de Gregorio, Ulderico Fugacci and Giovanni Petri.
The main focus of interest is the following: within topological data analysis, a well-developed pipeline exists to extract from data a topological summary, in the form of a persistence module, or its numerical description such as the barcode or persistence diagram. However, this description misses an important piece of the puzzle; homology essentially describes a pattern of obstructions to connectivity, and the existence of non zero homology classes points to the presence of such obstructions.
The very nature of homology, however, is that of an equivalence class of cycles within a simplicial structure obtained from the data. As humans, instead, we build intuition about the data at hand based on the *actual cycle*, and not equivalence classes of those. In other words, the process of computing a quotient by which homology is constructed identifies cycles on the basis of their topological relations, but in doing so mixes together a potentially very large number of objects that, depending on the context, could instead convey very different meaning.
This problem is sometimes called *localization of homology*. It essentially boils down to considering the preimage of the quotient map

$$[\cdot] \,:\, Z_k \twoheadrightarrow H_k$$

that associates to a cycle $c \in \ker \partial_k$ its homology class $[c] \in H_k$. The cycles corresponding to a homological equivalence class are called *representatives*, or *representative cycles*. Working with vector spaces, we can equally regard the above as a map from the k-chains into homology, and therefore stating the problem as follows: how can we choose *good* representatives of homology classes? That is, good (in some sense) k-chains to *depict* each homology class as an actual subset of the simplicial complex?

The problem has received attention in recent years. It is clear that not much interest can be devoted to the zero-dimensional case, as representatives of connected components are straightforward.

Aside from that, one first criterion that comes to mind when tackling this problem is obtaining a set of representatives that achieves some sort of geometric minimality. For example, for geometric simplicial complex one can associate to each cycle a measure, like an area for cycles in the plane, or a volume for cavities in space. Even when lacking an explicit embedding of the complex in the space, one can always consider the pairwise dissimilarities between vertices, and associate to each cycle a notion of *length*. This is possible even when lacking a dissimilarity measure, by just considering the hop distance on a graph. Finally, without requiring knowledge of an embedding, but with the request that our data forms a metric space, approaches have emerged that try to leverage the concept of minimum spanning tree to build representatives of "small" size.

The computation of a *minimal "basis"* (where by basis we mean a set of cycles whose homology classes span the whole homology vector space) has been proven to be a complicated problem. When considering dimension 2 and above, it has been shown by Chen and Freedman that the problem is NP-complete ([15]). The case remained open for dimension 1, until polynomial algorithms started to appear for the task. This is the avenue that we have followed in the construction of our minimal scaffold.

Another approach that has been employed for the task of choosing generators in a non-arbitrary manner concerns the case when data falls under the hypotheses of Alexander duality. This result provides an isomorphism between (co)homology in dimension 0 and $n-1$; since, as already mentioned, choosing representatives in dimension zero is trivial, this provides a principled solution for this particular case.

In the remainder of the chapter we will consider this problem, focusing especially on the case of the bases of minimal length, while also touching on some of the other approaches. We have begun our work by putting together a working implementation of an algorithm that computes a minimal homology basis, inspired by an applicative question: representative cycles of (persistent) homology classes had found fruitful use in data analysis, but could we provide a principled choice for said representatives?

3.1 Network skeletonization via Minimal Homology Bases

The question from which the minimal scaffold project stemmed was rooted in the application of complex network theory to neuroscience.
Network science has long represented the cornerstone theory in dealing with complex, heterogeneous multi-agent systems. Network descriptions have found wide applications and had a significant impact on a wide range of fields ([16, 17]), including social networks ([18, 19]), epidemiology ([20, 21]), biology ([22, 23]), and neuroscience ([24–26]).

Recently, new approaches to the analysis of networks and, more generally, complex interacting systems have emerged which leverage topological techniques ([27–30]), and most notably persistent homology.

Indeed, the range of fields into which TDA has found applications spans material science ([31, 32]), biology and chemistry ([33–39]), sensor networks ([40]), cosmology ([41]), medicine and neuroscience ([42–50]), manufacturing and engineering ([51–53]), social sciences ([54, 55]), computer vision ([56–58]), and network science itself ([59–65]).
The theory of persistence has recently been proposed as a framework for the topological *skeletonization* of spaces, particularly weighted graphs and networks ([66–69]). In [47], the generators of persistent homology are used to build one instance of network skeletonization called *homological scaffold*. This method has a serious drawback, consisting in the large degree of arbitrariness in the choice of one representative cycle from the many equivalent generating cycles of the same homology class. This pitfall is a direct consequence of the homology classes being equivalence classes, and it affects any attempt to localize cycles ([70, 49]). Minimal homology bases, as remarked above, have been investigated in the literature ([71, 72]), but a real breakthrough has only come thanks to the introduction of the first efficient algorithm for the computation of bases in dimension one ([73]). Here, we set out to address the issue of giving a principled definition of the scaffold by searching for a form of *canonicity* in the choice of generators, namely by computing *minimal representatives* of homology bases.
Next, we leverage said minimal bases to propose a new approach to network skeletonization, the *minimal scaffold*, which largely overcomes the limitation of the

previous one. While the minimal scaffold is not unique in the most general case possible, we provide strong guarantees and caveats on when and to what degree it is well-defined. This constitutes the main methodological contribution of this section. Then, we capitalize on the properties of the novel framework: as remarked in the chapter's introduction, having reliable representatives when dealing with neuroscience data can provide information that is actually interpretable by an expert of the field. We will showcase as application an analysis of this type.
We finally conclude the section by addressing one last question. The construction proposed here is provably more reliable than the previous, loose one, but this comes at a significant computational cost. In the light of this we foresee that it could be of value to verify whether, in some circumstance, the two constructions (the minimal and the loose scaffolds) are statistically related, so that the easy-to-compute one can be approximately used as a proxy of the minimal one. We provide heuristic evidence that, for a range of popular random models, the two objects are sufficiently well-related.

3.1.1 The homological scaffold

The homological scaffold originated from the intuition that traditional, graph-theoretical tools in network analysis were naturally able to capture significant properties of a network ([74]), but proved not as effective in detecting multi-agent and large-scale interactions. Interest in searching for alternative descriptors of network relations arose, and soon works were published which leveraged invariants offered by computational topology ([75, 29, 28]).
In proposing the scaffold ([47]), the authors pointed out that the homological description is potentially able to summarize the network's *mesoscale* structure, i.e., features existing at a scale in between the purely local connections and the global statistic, to which previous methodologies were blind.
Furthermore, this structure could be analyzed over the continuous, full range of interaction intensities, without the need for ad-hoc domain-specific thresholds.
Homological cycles intuitively describe obstruction patterns. The presence of non-trivial homology within a given region of a network highlights its structure as non-contractible, forcing signals to flow over constrained channels, which in turn play the role of bridges.

To test this intuition, the homological scaffold was computed from resting-state fMRI data for 15 healthy volunteers who were either infused with placebo or psilocybin: the scaffold discriminated the two groups, as well as providing meaningful insight as to the impact of the psychoactive substance onto the pattern of information flow in the brain [47].

Consider a non-negatively weighted finite graph $W = (V, E, w)$ where w is a weight function on the edges $w : E \mapsto \mathbb{R}^+$), and let $\mathcal{F}$ be a filtration of simplicial complexes over the reals, constructed as follows: to each $\varepsilon \in \mathbb{R}$, consider the graph $W^\varepsilon := (V, E^\varepsilon)$, where E^ε is the set of edges $e \in E^\varepsilon$ if and only of $w(e) \leq \varepsilon$. Then, $\mathcal{F}(\varepsilon)$ is the flag complex of W^ε, and the maps between them are the obvious inclusions.
Notice, furthermore, that as we deal with finite point clouds, it must hold that the image $\mathrm{Im}\ w \subset \mathbb{R}$ is a finite set. This implies the filtration can equally be considered as indexed on a finite poset of the form $([n], \leq)$, with the bound $n \leq |E|$; consequently, we can write $\mathcal{F} = \{K^{\varepsilon_k}\}_{k=1}^n$.
We can now consider $PH_1 = H_1\mathcal{F}$ of this weighted graph, that is a persistence module in dimension 1. The finiteness hypothesis guarantees that the module is tame.
Let $\{b_i\}$ be a linearly independent set of generator cycles of $PH_1(W)$, as a module. That is, consider a set of 1-chains that belong to Z_1, that form a linearly independent system, and whose linear span in $PH_1(W)$ is the whole persistence module.
Since we are over $\mathbb{Z}_2$, each of the b_i's is completely identified by its support, which is a set of edges of E. In particular, we can depict set $\{b_i\}$ as a matrix whose rows are indexed by E and having the b_i's as columns. The sums of the rows, considered as natural numbers as opposed to $\mathbb{Z}_2$, form a new weighting function on the edges of W, the new weights counting precisely in how many persistent cycles an edge appears along the filtration.

Definition 54. (Homological Scaffold) Suppose W and $\mathcal{F}$ as above, and consider a linearly independent set $\{b_i\}$ of 1-dimensional generator cycles of the persistent homology. Consider the function $h_W : E \mapsto \mathbb{R}^+$

$$h_W := \sum_i \mathbb{1}_{e \in b_i} \tag{3.1}$$

where by $\mathbb{1}_{e \in b_i}$ we denote the indicator function $E \mapsto \mathbb{R}^+$ such that $\mathbb{1}_{e \in b_i}(e') = 1$ if e' appears in b_i, and 0 otherwise.

Then the **homological scaffold** of W is the weighted graph $\mathcal{H}(W)$ such that

- its vertex set coincides with the vertex set of W
- its edge set $E_{\mathcal{H}}$ is a subset of the edge set of W, consisting of edges with nonzero value for h_W
- its weight function is the restriction of h_W to $E_{\mathcal{H}}$.

In accordance with the above definition, building the homological scaffold of a weighted network W is a method of *network compression* or *skeletonization*. The definition also implies that edge weights are assigned by the number of basis cycles an edge belongs to.

We provide an example, referring to Fig. 3.1. In panel (a), a filtration of simplicial complexes arising from a point cloud is depicted. At each step, highlighted in purple is a representative of a persistent cycle (i.e. of a bar in the barcode), each at the scale at which it is born.
In panel (b), the corresponding homological scaffold is represented: it amounts to taking the union of the cycles of panel (a), i.e. stacking generators of PH_1, each contributing unitary weight.

In the following, we shall sometimes refer to the homological scaffold as the *loose*, or *original* scaffold, to contrast it with the new definition of scaffold to follow.

As anticipated in the introduction, it is clear that there is a substantial source of arbitrariness in this definition.
Several different representative cycles exist which form a basis of the persistent homology (as a consequence of several different cycles belonging to the same homology class), and hence one must make a choice. For example, Fig. 3.2(a) depicts one specific cycle whose homology class generates (part of) the persistent homology group of the point cloud. At the same time, any other choice of edges forming a cycle around the same hole is homologically equivalent and, in principle, a legitimate choice for the set $\{b_i\}$.
In the original paper, the authors resorted to using the cycles as output by the *JavaPlex* implementation ([76]) of the persistent homology algorithm (based on the

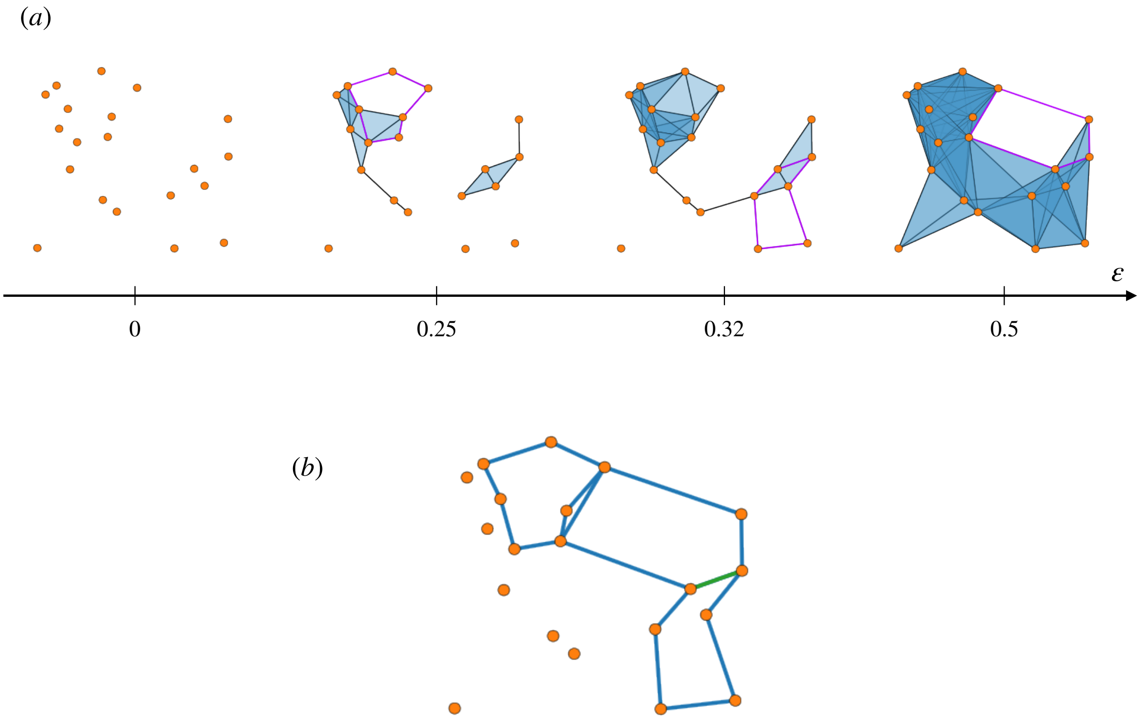


Fig. 3.1 (a) A point cloud in $[0,1]^2$ and the generators of PH_1, plotted on the filtration step they appear at (scale reported on the axis below). (b) The resulting homological scaffold. Edges in blue have weight 1, each belonging to only one generator. The edge in green has weight 2, as it belongs to two generators.

original implementation of [77]), and a posteriori checked the selected cycles for consistency. However, in principle, this means that the same simplicial complex written with two different orderings of the simplices could lead to different choices of generators, and therefore, to different scaffolds.

As such, we must be careful in the choice of nodes and edges output by the algorithm; while the presence of a generator denotes undeniably that an obstruction pattern exists, we cannot be as confident about its precise location in the network or the constituents that provide bridges around it. The homological scaffold defined in this way introduces noise in the localization of mesoscale patterns onto individual nodes and edges, a process which, if accurate, could provide valuable insight as to the functional role of single players in a network.

In this work, we try to work around the problem of cycle choice and give a stricter definition, by requiring that, among all possible representatives, those of *minimal total length* are chosen (e.g., Fig. 3.2(b)).

The original algorithm reported a computational complexity of the order $O(n^3)$ to obtain representatives of basis cycles.

(*a*) (*b*)

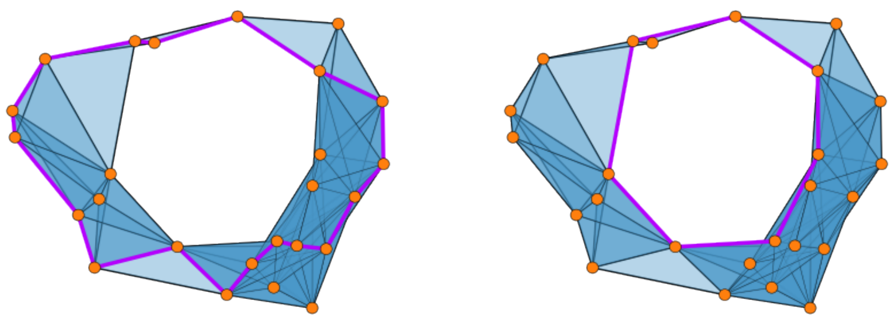


Fig. 3.2 A simplicial complex K with $\dim H_1(K) = 1$. Its homological scaffold (on a subset of the filtration steps, for clarity) is reported in panel (a): the chosen generator meanders around the hole. Furthermore, a different ordering of the list of simplices fed to the algorithm could return a different cycle. In panel (b), the shortest representative cycle is chosen: this choice is stable with respect to any ordering of the input, while at the same time endowing the generator with some metric and geometric meaning.

3.1.2 Minimal Bases

The search for minimality in the computation of the scaffold was made feasible by the introduction of efficient algorithms to compute the minimal representatives of a homology bases in dimension one.
It is known that in dimension higher than one, minimal representatives of a homology basis will remain elusive. Indeed, Chen and Freedman ([15]) proved that the problem of obtaining these minimal representatives is computationally intractable, being at least as hard as the notoriously NP-Hard Nearest Codeword Problem. Furthermore, it is even NP-Hard to approximate within any constant factor, meaning that no polynomial-time algorithm exists to obtain an approximate minimal basis that differs from the exact one by at most a multiplicative constant. In the light of this, we must necessarily restrict our attention to the 1-dimensional case, i.e., computing minimal representatives of a basis of H_1.

Given a simplicial complex K, let us consider $C_1(K,\mathbb{F})$ the vector space generated by the 1-simplices of K and Z_1 the vector space of 1-cycles, i.e., $Z_1 = \ker\partial_1$. Given a 1-cycle $b \in Z_1$, let $\mu(b)$ be its length, i.e., the sum of the weights of the edges that form it

$$\mu(b) = \sum_{e\in b} w(e)$$

Denote by $[b]$ the homology class of b. Finally, let $\beta_1 := \dim H_1(K)$ be the first Betti number of K.

We want to obtain a set of β_1 1-cycles $\in Z_1$

$$\{b_1,\ldots,b_{\beta_1}\} = \underset{\mathrm{Span}\{[b_i]\}=H_1}{\operatorname{argmin}} \sum_i \mu(b_i) \tag{3.2}$$

that is, a set of cycles of minimal length whose homology classes span $H_1(K)$. In accordance with the literature, we call this set a *minimal homology basis*, with a slight abuse of terminology, as it would be more appropriate to call it a *minimally-represented homology basis*.
In 2018, Dey et al. ([73]) introduced a polynomial-time algorithm to obtain said representatives. Building on the work of Horton ([78]), de Pina ([79]), and Mehlhorn

et al. ([80]), the algorithm sets off to compute a basis of the space of cycles. Then, it applies a cohomological technique called *simplex annotation* ([81]) to lift a basis of cycles to a basis of the homology group H_1, while at the same time enforcing the minimal length constraint. A sketch of the algorithm follows.
ALGORITHM: MINBASIS(K)

- A basis of the cycles group Z_1 is found via a spanning tree. Each edge in the complement of the spanning tree identifies a candidate cycle ([78]).
- An annotation of the edges is computed via matrix reduction ([81]). This yields the dimension β_1 of H_1, as well as an efficient tool to determine if two cycles b_1 and b_2 are linearly dependent in H_1 ($[b_1] = [b_2]$).
- A set of *support vectors* is generated which maintains a basis of the orthogonal complement in H_1 of the minimal basis cycles.
- Iteratively for each dimension of H_1, the candidate set of cycles is parsed in search of cycles b's that are linearly independent in homology from the previous ones (exploiting the support vectors). Among these, the μ-shortest one is added to the minimal basis.
- The set of support vectors is updated for the remaining dimensions to enforce it remain a basis of the orthogonal complement of the basis.
- The last two steps above are repeated until completion of the minimal basis.

Call $B = \{b_i\}$ the output of MINBASIS on input K.
Theorem (3.1, [73]) Cycles in B form a minimal homology basis of $H_1(K)$.

Notice that the minimal homology basis is guaranteed to exist, as we only work with finite simplicial complexes, which imply the existence of a finite number of bases. However, it needs not, in general, be unique. Several different cycles of the same minimal length may all belong to the same homology class of a basis cycle. Heuristically, this is especially true in case the input complex is unweighted (equivalently, has equal weights for every edge), in which case the length of a cycle is the number of edges that form it. Furthermore, there exist cases when different sets of cycles of minimal length generate the same homology space, and are not even pairwise homologous.

We will treat the problem of the uniqueness of the minimal basis in more detail in the following, and account for it explicitly in the construction of the minimal scaffold.

The computational complexity of the above procedure is evaluated ([73]) to $O(n^2\beta_1 + n^\omega)$ where n is the number of simplices in K and ω is the fast matrix multiplication exponent, which as of 2014 is bounded by 2.37 ([73, 82, 83]). This yields a worst-case complexity of $O(n^3)$ in the number of simplices for general complexes, which we recall is itself of order 3 in the number of points in the worst case.

3.1.3 Minimal Scaffold

In this section, we introduce an alternative definition for the homological scaffold, which we call minimal, based on the minimal representatives obtained above, which aims at overcoming the arbitrariness in the cycle choice of the previous definition. After addressing the simplest case, we analyze its uniqueness properties and introduce a second, more refined, definition.

Let $\mathcal{F}$ be the filtration of simplicial complexes induced by a non-negatively weighted finite graph W. For all filtration steps ε, define, as per (3.2), $B^\varepsilon := \{b_i^\varepsilon\}$ the minimal homology basis of $H_1(K^\varepsilon)$. Take the disjoint union of minimal bases for ε varying on all filtration steps

$$B^* := \bigsqcup_\varepsilon B^\varepsilon$$

Definition 55. Suppose W, $\mathcal{F}$ and B^* as above. Similarly to the loose case, define the function $h_{W,min} : E \mapsto \mathbb{R}^+$ as

$$h_{W,min} := \sum_{b \in B^*} \mathbb{1}_{e \in b} \tag{3.3}$$

Then, we define the **minimal scaffold** of W as the weighted graph $\mathcal{H}_{min}(W)$ whose:

- vertex set coincides with the vertex set of W
- edge set E_m is a subset of the edge set of W, consisting of edges with nonzero value for $h_{W,min}$
- weight function is the restriction of $h_{W,min}$ to E_m.

The minimal scaffold amounts, again, to the stacking of generator cycles across a filtration. However, two differences are to be noted with respect to the loose definition. First, we require the representative cycles to be minimal. Second, we point out that while the loose scaffold is built by aggregating the generator cycles of $PH_1(\mathcal{F})$, the minimal scaffold is built by independently computing a minimal basis for each $H_1(K^\varepsilon)$, for all ε. Notice that, since cycles are modified throughout a filtration, it would be meaningless to talk about a minimal representative over a certain persistence interval. This also means that its computation can be effectively parallelized by assigning different filtration steps to different jobs, and later recombining the outputs.
An interesting phenomenon that descends directly from the above peculiarity is that the minimal scaffold of random point clouds tends to display a more pronounced triangular structure (clustering) around cycles. Indeed, as longer (or, in non-metric filtrations, later) edges are introduced, a cycle can be shortened (by the triangular inequality) by a longer edge which cuts a corner. Since at each step the algorithm records the minimal representative, upon aggregating the minimal scaffold one finds each cycle in its progressively shorter version, and the *history* of the shortening is visible as a padding of triangles around it.

Considering the example of Fig. 3.3, in panel (a) we observe an example of a filtration of simplicial complexes. At each step, highlighted in purple we may see the minimal representative of a homology class, together with its evolution history. At filtration value 0.26, we observe a pentagon being reduced to a shorter, quadrilateral cycle by the addition of a longer edge. This is an example of the phenomenon explained above. Fig. 3.2 gives a visual description of the difference between a minimal and generic cycle.
The union of these progressively shorter cycles for all steps (weighted according to Definition 3.3) is the minimal scaffold, as seen in Fig. 3.3 panel (b).

We remark that, if there is no ambiguity in the construction of a filtration of simplicial complexes from a point cloud, or from a weighted graph, we will indifferently speak of the scaffold as a function of either of them ($\mathcal{H}_{min}(C)$, or $\mathcal{H}_{min}(W)$, or $\mathcal{H}_{min}(\mathcal{F})$).

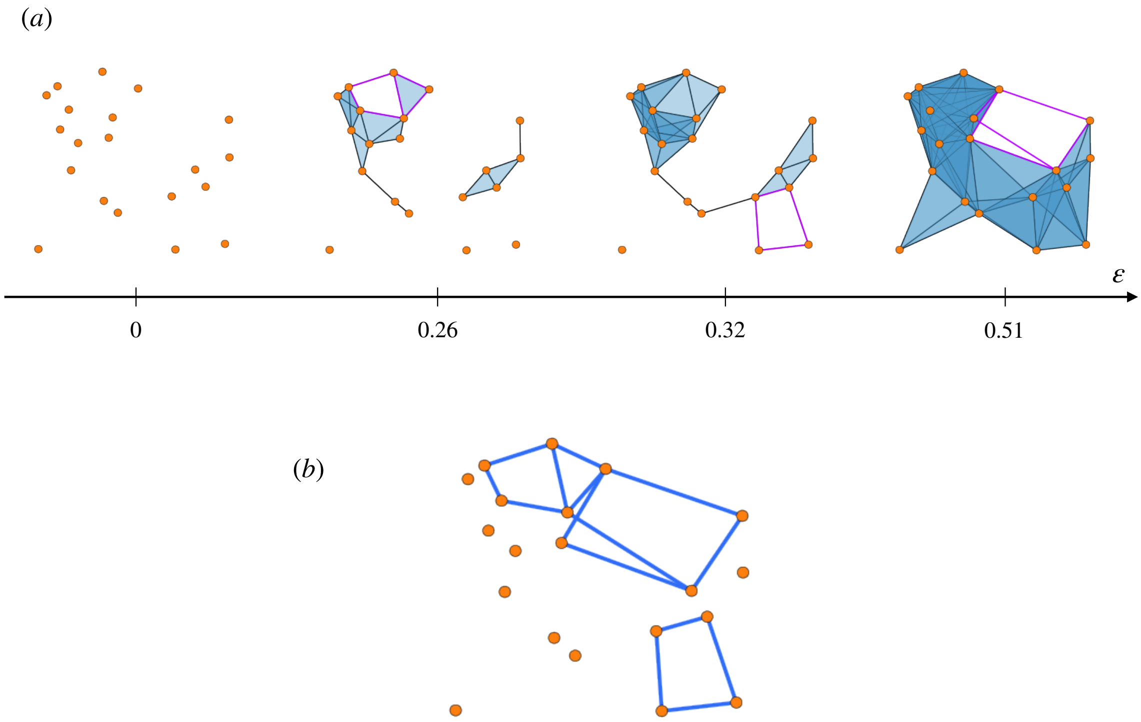


Fig. 3.3 (a) The same point cloud of Fig. 3.1. Along the filtration we show the evolution of minimal generators, which can get progressively shorter as new edges are introduced. For example, at $\varepsilon = 0.26$, the pentagonal cycle gets cut to a shorter quadrilateral, albeit with an individual longer edge. This evolution is accounted for in the minimal scaffold, which displays the triangle-rich structure mentioned above. (b) The resulting minimal scaffold (weights not reported).

We have mentioned that the scaffold amounts to a change in weighting in the input graph

$$h_{W,min} \; : \; E \mapsto \mathbb{R}^+$$

altering the original weights of the edges. Additionally, considering *node strength* (i.e. the sum of the weights of the edges incident to a given node), it can equally be considered as a function

$$\mathcal{H}_{min} \; : \; V \mapsto \mathbb{R}^+$$

assigning weights to nodes. Considering the reliability of the choice of edges in the procedure, this explains why the minimal scaffold can be utilized to associate mesoscopic features with single nodes and links.

Computational Complexity

For large input sizes, the cost of assembling the minimal basis cycles into the scaffold is negligible with respect to the cost of computing such minimal basis.

We know that each run of Dey's algorithm costs $O(|K|^3)$ in the worst case ([73]), and in the worst case $|K|$ is itself $O(n^3)$ where n is the number of points.
The number of filtration steps has an upper bound of $O(n^2)$ (i.e., the number of edges) in the worst case, as in general every edge may carry a different weight. Hence Dey's algorithm has to be run once for each edge in the worst case.
This yields a theoretical worst-case complexity of order $O(n^9 n^2) = O(n^{11})$. Therefore, while the minimal scaffold is undeniably a polynomial-time algorithm, its practical computation is often hindered by its dire lack of scalability, especially if compared against the loose version, which has a far more favourable complexity.
A comparison of running times is carried out in Fig. 3.4, which clearly shows that computing the minimal scaffold on an ordinary machine can quickly become troublesome.

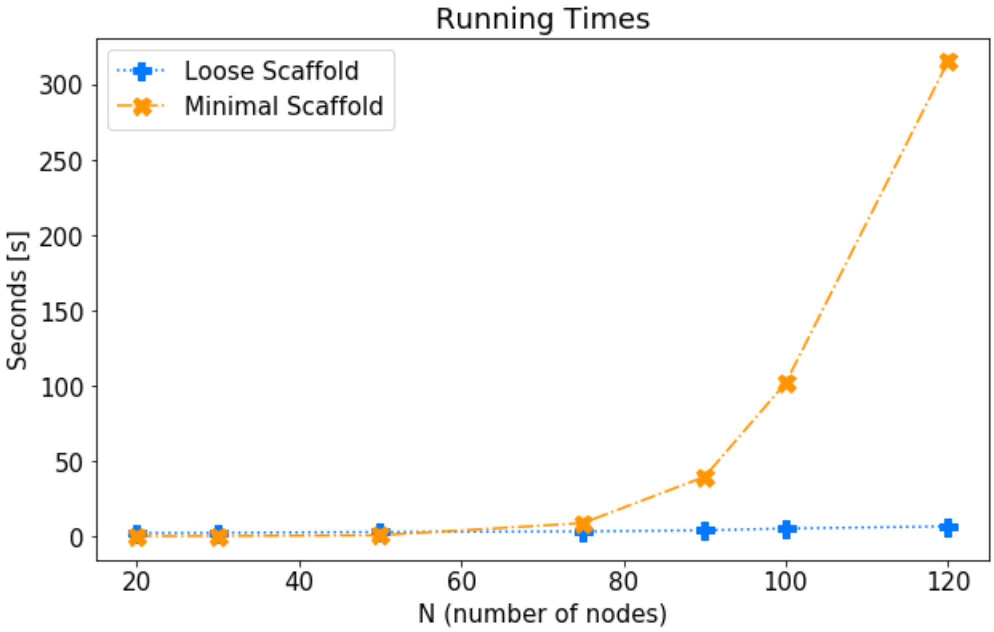


Fig. 3.4 The running times of computing the minimal and loose scaffolds for Watts-Strogatz weighted random graphs. For all instances, number of nodes N is indicated on the x-axis. Number of stubs k is $N/2$, and rewiring probability is $p = 0.025$.

3.1.4 Uniqueness of the minimal scaffold

The uniqueness of the minimal scaffold depends on the uniqueness of the minimal basis. Indeed, if there exists only one possible set B^* of cycles forming a minimal basis, then the scaffold is uniquely determined. Two issues affect the uniqueness of set B^*.

Draws

The first one arises when two or more different and homologous basis cycles are of the same minimal length. This case is relatively simple to work around: we modify the definition of minimal scaffold to keep track of all variants of minimal basis cycles, dividing the weight equally among them.
Specifically, to account for this issue we have slightly modified Dey's algorithm. In its last step described above, one is concerned with finding all cycles whose annotation is not orthogonal to the given support vector: among these, the one with minimal length is chosen as a basis cycle. Instead, we keep track of *all* such cycles

with the same minimal length. This does not alter the complexity, as one needs to check all possible cycles anyway. We call this case a *draw*.

Therefore, we modify set B to become a set of *sets of cycles*. Given complex K, we define a minimal basis *with draws*

$$\tilde{B} := \bigcup_{i=1}^{\beta_1(K)} \{b_{i,1}, ..., b_{i,n_i}\}$$

where for all $i = 1, ..., \beta_1(K)$, the cycles $b_{i,j}$ with $j = 1, ..., n_i$ are homologous and have the same minimal length. Furthermore, for every choice of $j_i \in \{1, ..., n_i\}$, $\mathrm{Span}_i\{b_{i,j_i}\} = H_1(K)$. Call $V_i := \{b_{i,1}, ..., b_{i,n_i}\}$ each set of draws, i.e., *variants* of the i^{th} minimal basis cycle, $\forall i = 1, ..., \beta_1(K)$.
In the example of Fig. 3.5(a) and (b), we have set $\tilde{B} = \{\ \{b_{1,1}, b_{1,2}\}\ \}$, whereas set B might have indifferently been equal to $\{b_{1,1}\}$ or to $\{b_{1,2}\}$, whichever happened to come first in the search.
The minimal scaffold is modified accordingly. Given the usual filtration $\mathcal{F}$, let $\tilde{B}^{\varepsilon}$ be the minimal basis with draws of $H_1(K^{\varepsilon})$. Again, we aggregate all variants of minimal basis cycles along the filtration

$$\tilde{B}^* := \bigsqcup_{\varepsilon} \tilde{B}^{\varepsilon}$$

Then, we define the weighting function *with draws* $\tilde{h}_{W,min} : E \mapsto R^+$

$$\tilde{h}_{W,min} := \sum_{V \subset \tilde{B}^*} \frac{1}{|V|} \sum_{b \in V} \mathbb{1}_{e \in b} \tag{3.4}$$

and the resulting *minimal scaffold with draws* $\tilde{\mathcal{H}}_{min}(W)$ is built from $\tilde{h}_{W,min}$ as in Definition 3.3.

The meaning of the above definition is that all variants of all minimal basis cycles are taken into account when building the scaffold, and the weights are assigned dividing each variant's contribution by its cardinality, for each filtration step. In the example of Fig. 3.5(c), the two cycles forming the variant of the only generator are multiplied by a factor of $\frac{1}{2}$ and then summed: therefore, common edges outside the diamond are assigned weight 1, consistently with the minimal scaffold in definition (3.3), whereas the four edges forming the perimeter of the diamond each get assigned weight $\frac{1}{2}$.

With the introduction of draws, we settle the case when ambiguity arises among individual cycles, without interactions. As an example, we can state the following result.

Proposition If $\mathcal{F}$ is such that, for all ε in the filtration, each basis cycle belongs to a different connected component of K^{ε}, then the minimal scaffold with draws $\tilde{H}_{min}(\mathcal{F})$ is unique.

Pathological cases

The other issue arises when there exist sets of minimal cycles that are not linearly independent. Suppose that three different cycles generate a homology group of dimension two, i.e., when three minimal cycles are pairwise independent in homology, but threewise dependent. In this case, two generators are sufficient to span H_1 and, if their lengths are arranged pathologically, there is no principled way to choose two out of the three.

Suppose for example that three cycles b_1, b_2 and b_3 are such that

$$\mu(b_1) < \mu(b_2) = \mu(b_3) \text{ and } [b_1] = [b_2] + [b_3]$$

In this case, both bases $\{b_1, b_2\}$ and $\{b_1, b_3\}$ span the same homology space, and are of equal minimal length. The minimality criterion fails in this case.

One could believe that such a configuration can only happen in the most general spaces, and that by imposing some mild hypotheses on the input data one could rule the pathology out. In fact the opposite is true, this degeneracy being possible even after enforcing very strong conditions on the data.

Counterexample Even if W is planar and an isometric embedding $W \hookrightarrow \mathbb{R}^2$ exists (i.e., the input planar weighted graph can be accurately drawn onto the plane), the minimal scaffold $\tilde{H}_{min}(W)$ needs not be unique.

In fact, consider complex K arising from the geometric, planar graph in Fig. 3.5(d). Its homology $H_1(K)$ is generated by two cycles; possible generators are depicted in Fig. 3.5(e). Since the outer cycle b_1 is the shortest, and the two inner ones b_2 and b_3 are of equal length, the minimality criterion can not solve between $\{b_1, b_2\}$ and $\{b_1, b_3\}$, as both are acceptable minimal bases. The minimal scaffold (with or without draws) is not unique in this case.

Clearly, the same could happen with more than three cycles, with a larger number of possibly ambiguous configuration. Therefore, if we allow for a high degree of symmetry in the input, this pathology could arise even in the rather tame context of planar graphs on $\mathbb{R}^2$. This issue is rather delicate, in the sense that not only the algorithm is unable to make a principled choice; it is not even capable of detecting when such a configuration takes place. In fact, this is more of a feature of homology than a flaw in the skeletonization framework: what our eyes see as different cycles are in fact homologically equivalent, and it is impossible to use homology to tell them apart.

We however remark that, for complexes arising from real-world data, this type of configuration is actually pathological. Indeed, the following generality result holds

Proposition Assume a point cloud $C = \{X_i\}$ such that $X_i \sim U([0,1]^d)$ independently. Then, almost surely, the minimal scaffold $\mathcal{H}_{min}(W)$ (either with or without draws) is unique.

If the input point cloud is sampled uniformly at random in some $\mathbb{R}^d$, then edge lengths are distributed according to an absolutely continuous probability law. Therefore, given two edges e_1 and e_2, $\mathbb{P}[\mu(e_1) = \mu(e_2)] = 0$. The same holds for any two non-identical cycles, and any two homology bases (being but finite sets of edges): the probability of them sharing the exact same length is zero. By finiteness of the input, at least one minimal homology basis exists and, by the above reasoning, almost surely this basis is unique for each filtration step. Then, with probability 1 the minimal scaffold is unique.

This result is actually quite general: whenever we can assume our input data to be subject to noise, then we are in principle allowed to rule out pathological same-length cycles. In these cases, the minimal scaffold is unique.

We remark that this uniqueness result is compatible with the phenomenon of the concentration of measure: while for a very high-dimensional space or a very large number of points we know from theory that the distribution of length of edges concentrates towards its mean value, the probability of two edges (and hence two cycles) having the same length is still zero. One needs to be careful, however, that the probability of two cycles differing in length by less than some $\varepsilon > 0$ could grow very rapidly with ε.

In summary, the minimal scaffold with draws $\tilde{\mathcal{H}}_{min}$ is well-defined up to some pathological circumstances, where it may depend on the ordering of the input.

Implementation

Alessandro de Gregorio and myself have built an implementation of the computational pipeline above, including Dey's algorithm and libraries to construct the minimal scaffold, with or without draws. The code is freely available on GitHub at [84], with some usage examples. It is built to allow both serial and shared-memory, multi-threaded parallelism across the filtration steps to improve efficiency. It can significantly benefit from a parallel execution, but can as equally run on a vanilla, serial environment.

Another trick we suggest that can sometimes make a dramatic difference in computation time is the following: if there exist intervals along the filtration where no homology in dimension 1 appears, it is definitely worth it to exclude them from the computation, as the algorithm a priori constructs some expensive combinatorial structures before it can eventually realize that no homology exists. Therefore, it is recommended to obtain such information via a fast algorithm for persistent homology such as Ripser [85], and use it as a prior to expedite computations for the minimal scaffold.

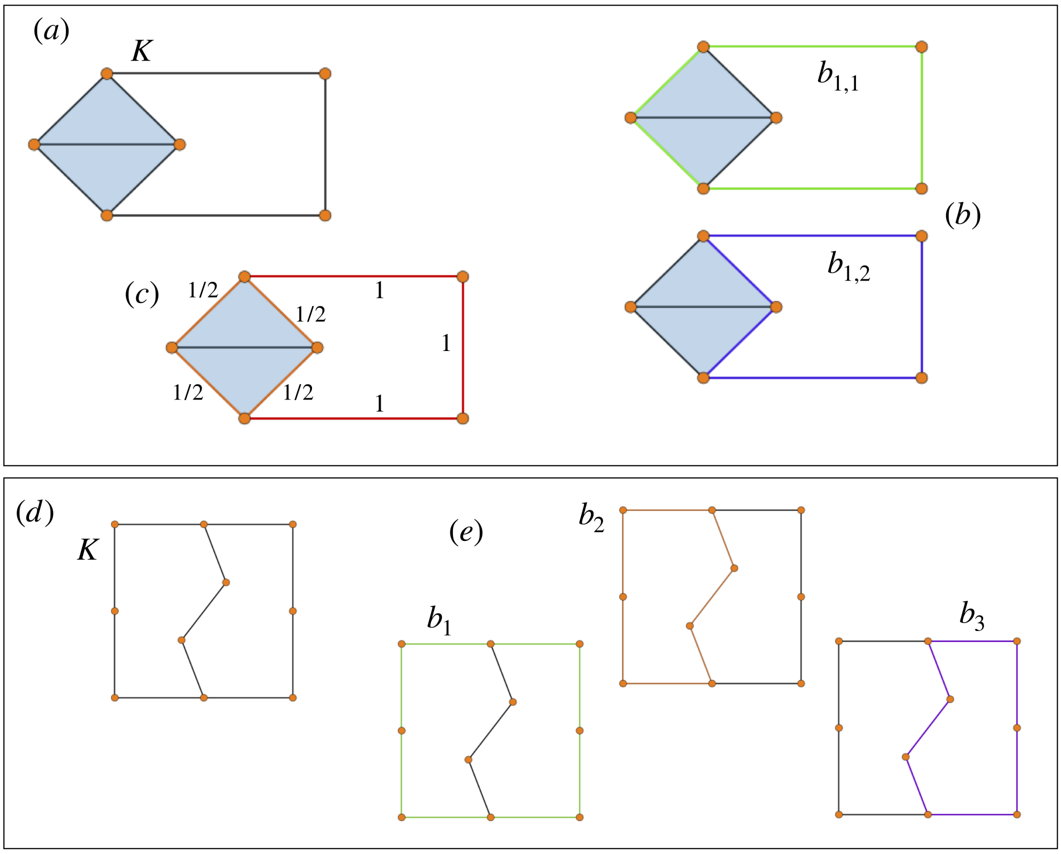


Fig. 3.5 Top panel: (a) A simplicial complex K. (b) Two homologous and equally minimal generators of $H_1(K)$. (c) The minimal scaffold with draws $\tilde{\mathcal{H}}_{min}(K)$. The weight is equally divided among the variants of the minimal representative. Bottom panel: (d) A simplicial complex K on the represented point cloud. $H_1(K)$ has dimension 2. (e) $\mu(b_1) < \mu(b_2) = \mu(b_3)$. A minimal basis can either be composed of $\{b_1, b_2\}$ or $\{b_1, b_3\}$, hence it is not unique.

3.1.5 Applications

As illustrative examples, we show here a few applications of the minimal scaffold. Through it, we obtain meaningful subsets of known networks in neuroscience, and rank their constituents by their “topological importance".

The C. Elegans dataset is a correlation network of neural activations of the nematode worm Caenorhabditis Elegans. C. Elegans has become a model organism due to the unique characteristic of each individual sharing the exact same nervous system structure.
The input consists of a symmetric weighted adjacency matrix over 297 nodes, each representing a neuron. Edge weights represent (quantized) time correlations between the firing of neurons, ranging from 1 to 70.
The minimal homological scaffold of its brain map highlights the *geometry* of the obstruction patterns, i.e., the precise areas where nervous stimuli are less likely to flow. We stress the improvement obtained by the minimal scaffold over the loose one, in that it is not only able to identify the *presence* of a “grey area" in the network, but it can as well provide a reliable boundary for it, and identify which neurons and inter-neuron links are responsible for information flowing around the obstruction.
As an interesting example, we see in Fig. 3.6 the top 25 neurons ranked in descending order of relative node strength (sum of weights of incident edges) with respect to the average node strength. We can identify four nodes, labeled 81, 260, 36, and 37, which hold a significantly higher relative strength than the rest. This implies their presence in many minimal cycles across several scales, hence suggesting that they play a crucial role in the fabric of information flow within the nematode’s brain.

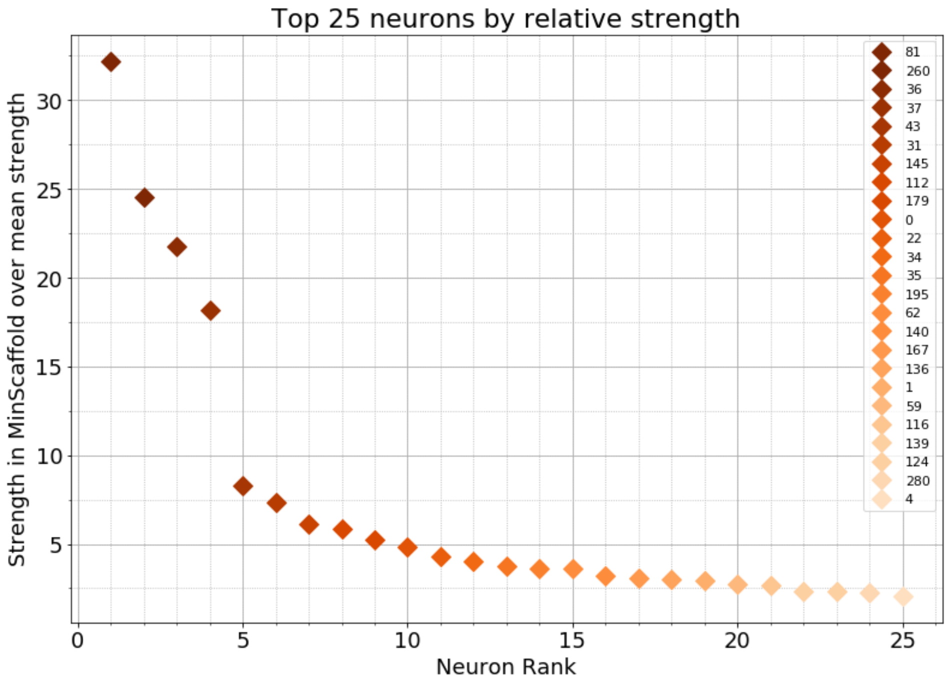


Fig. 3.6 The top 25 neurons by relative node strength in the minimal scaffold over average strength in C. Elegans (mean 36.41). Four neurons show a significantly higher relative strength than the others.

The same type of analysis was repeated on the correlation network of brain activities in an 88-parcel atlas of the human brain, obtained through fMRI imaging at resting state. The data is courtesy of the Human Connectome Project ([86]).
Again, the minimal scaffold identifies which regions and links in the human brain are key bridges for the flow of information. Two parcels stand out (Fig. 3.7(a)) as particularly relevant for network topology.
For a relatively small network such as this, we can visualize the scaffold as a proper subnetwork by a chord diagram (Fig. 3.7(b)), with edge weight represented by color intensity and node strength by the size and color of the vertex. We stress that, starting from a virtually complete graph over 88 nodes, we reduce the size from 3828 edges to just 191, while preserving the topological structure.
We can, as well, leverage libraries in computational neuroscience ([87]) to embed the scaffold in the actual human brain, with regions correctly located, projected on the three coordinated planes. In Fig. 3.7(c), for visualization purposes color intensities represent log-weight in the scaffold.
To better highlight the value of the scaffold in signalling brain network function, we constructed a suitable null model of the functional network, as was done in [88]. The technique consists in reshuffling the correlation matrix subject to the constraint of keeping a fixed spectrum, i.e. applying a random rotation, which guarantees the matrix remains positive semidefinite and hence a proper correlation matrix. An implementation of such a procedure can be found in [89].
The resulting randomized adjacency matrix is characterized by a vastly larger number of homological cycles than the original; so much so in fact that the computation of its minimal scaffold becomes cumbersome. However, even without computing them explicitly, we know for sure that the scaffolds of the original and randomized networks are totally different, specifically because they are built by aggregating two completely different persistence structures, i.e. the minimal scaffold does indeed highlight the functional information in the original dataset.

The possible applications in which the minimal scaffold could provide novel insight into the structure of brain data are many: any relatively small correlation matrix could be either compressed or its patterns analyzed, as is often the case in EEG [90, 48, 91, 50] or neuronal [44] studies, and in fMRI ones when using rather coarse atlases (e.g. [92, 93]).

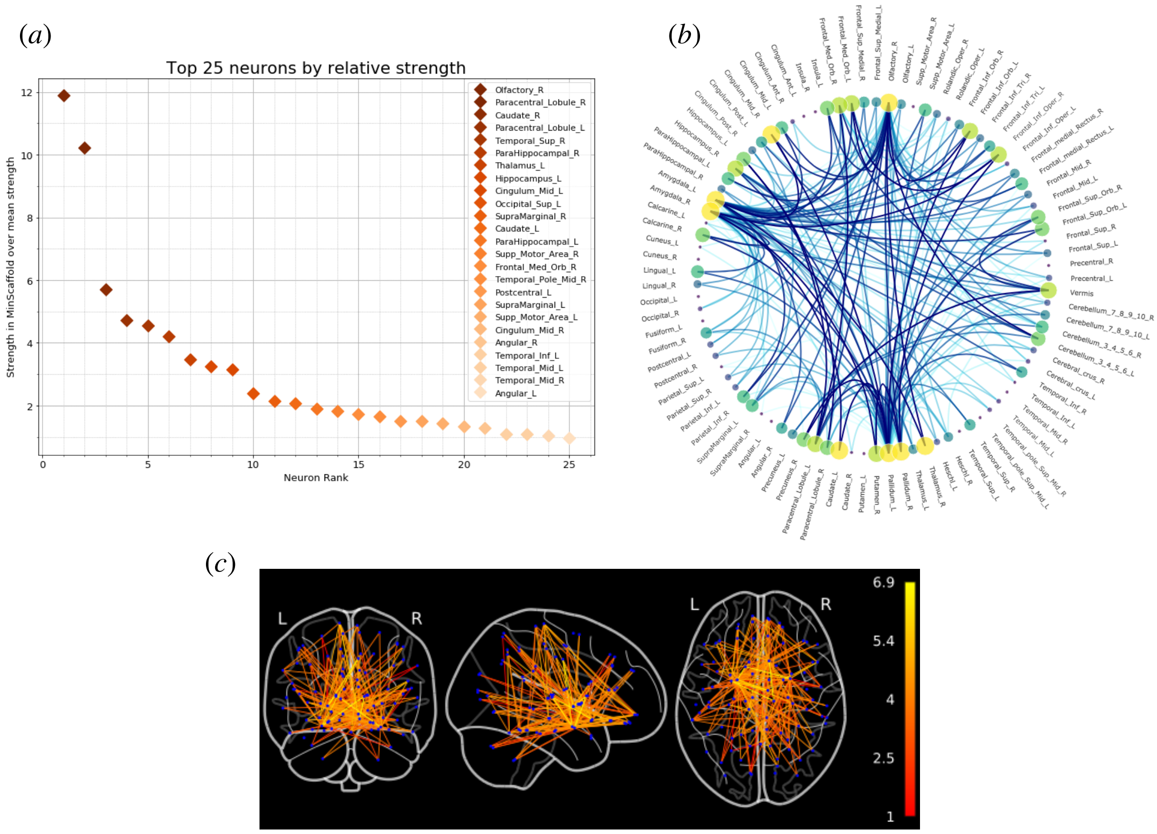


Fig. 3.7 (a) The top 25 brain regions in the human brain by relative node strength in the minimal scaffold over average strength (mean 546.7). Two neurons show significantly higher importance. (b) The chord diagram of the minimal scaffold. Node size represents node strength, edge color intensity represents weight in the scaffold. (c) The minimal scaffold embedded in the human brain, with regions accurately located, projected on the three coordinated planes. Edge color represents log-weight in the minimal scaffold (Log-scale for visualization purposes).

3.2 Comparison of Scaffolds

As the last contribution for this work, we consider a comparison between the minimal and loose scaffolds.

We have already pointed out that the minimal scaffold in general offers superior guarantees as a tool, both for network analysis and network skeletonization. On the other hand, the loose scaffold clearly has an advantage in terms of computational complexity: while it is in principle viable for most of the applications where persistent homology has been employed, the minimal scaffold, even adopting filtration-wise parallelization, requires a vastly larger amount of computational power, which effectively limits its range of application, unless run on dedicated, high-performance infrastructures.

A reasonable question to ask is the following. If one is interested not in the exact structure of the scaffold, but only in its statistical behaviour, could the loose scaffold provide a sufficient approximation of the minimal one? In a more concrete example, if instead of wondering exactly which nodes in a network are the most topologically important one is interested in the distribution of the degree sequence of the minimal scaffold, could the loose one come to one's help?

To answer this question, we have performed comparisons of several graph metrics in the two scaffolds of C. Elegans. Further, to gain insight into the general case, we have sampled two families of random graphs at different parameter values, one for geometric graphs (Random Geometric Graph), and one for non-geometric graphs (Weigthed Watts-Strogatz).

C. Elegans

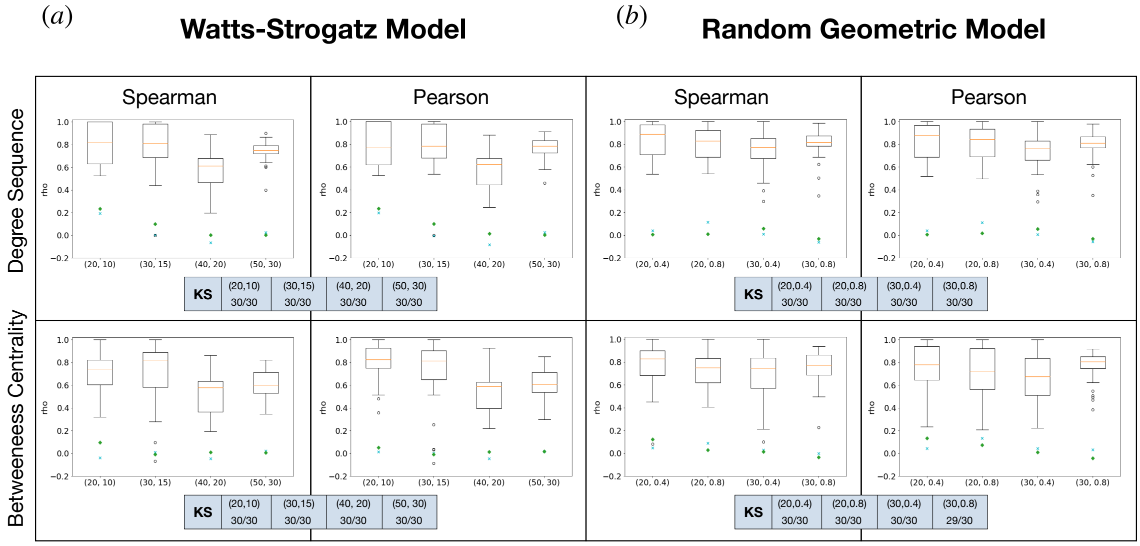
For the C. Elegans dataset, we have compared the following graph metrics of the minimal and loose scaffolds:

1. Degree Sequence
2. Node Strength
3. Betweeness Centrality
4. Closeness Centrality
5. Eigenvector Centrality
6. Clustering Coefficients
7. Edge weights

Results (reported in the Table of Fig. 3.8(c)) indicate that, for metrics 1 to 5, the two scaffolds are very well correlated. So for example the cheap, loose scaffold is a reliable proxy of the distribution of the “true" degree sequence (scatterplot in Fig. 3.8(d)).
We instead observe poor correlation of edge weights and clustering coefficients. The first one is not unexpected, since the edge weighting procedure is conceptually different in the two scaffolds: while in the minimal one we consider a different basis for each filtration step, the loose scaffold considers bases of the persistent homology space, drastically reducing the number of cycles considered. To make it clearer, in general set B^* has cardinality much larger than the dimension of PH_1. It is therefore explicable that the distributions of edge weights do not generally agree.
Clustering coefficients, on the other hand, are a measure of how “triangular" a graph is around a given node. As remarked in Section 3.1.3, another consequence of assembling the scaffold from the minimal bases of the H_1's is that a large number of artificial triangles appear around cycles. In this case too, therefore, the poor correlation is easily explained.

Random Graphs

Drawing inspiration from [65], we repeat the analysis on random graph samples. [65] divides random networks into two categories: those created from edge weighting schemes and those created from points in the Euclidean space. We have chosen to analyze the weighted Watts-Strogatz (WS) model as representative of the first class, and the geometric random model as representative of the second. We remark that weighting needs to be introduced in order to compute persistence; while for geometric graphs this simply requires computing the Euclidean distance, for the Watts-Strogatz model it requires an ad-hoc procedure that is described in detail in the supplemental material of [65].
We briefly recall that a WS graph is parametrized by the number of nodes, by the number of stubs to rewire, and by the rewiring probability. A random geometric graph is instead parametrized by the number of points to sample (uniformly) in $[0,1]^d$, and by a cutoff value that acts as distance threshold, beyond which no edge is introduced.
In both cases, we observe good agreement on key statistics, as reported in Fig. 3.8(a) and (b). Each bar is obtained by computing the correlation of the reported statistic on



(c)

	Spearman		Pearson	
Metric	Corr	p-val	Corr	p-val
Node Degree	0.953148	3.1842e-155	0.975559	3.4463e-196
Node Strength	0.772330	4.3712e-60	0.700653	3.7250e-45
Betw. Centrality	0.952098	7.7348e-154	0.986412	1.8813e-233
Closeness Centrality	0.921274	5.1143e-123	0.960413	8.7695e-166
Eigenvector Centrality	0.880711	9.5943e-98	0.858564	1.3911e-87
Clustering Coefficients	0.412889	1.1778e-13	0.358577	1.9337e-10
Edge Weights	0.226321	1.3586e-09	0.086226	0.0224

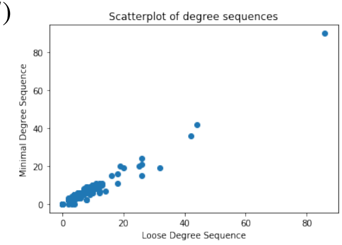


Fig. 3.8 Correlations between the minimal and loose scaffold. (a) Comparison in the weighted Watts-Strogatz model. Degree sequence and betweenness centrality in the two scaffolds are compared, using Pearson and Spearman correlation coefficients. Each box is computed over a sample of 30 weighted Watts-Strogatz random graphs, with parameters as reported on the x-axis: the pair (N,k) indicates a WS model on N nodes, with k stubs to rewire. The rewiring probability is 0.025. The cyan crosses and the green diamonds represent the average correlation value against the loose and minimal null models, respectively. (b) Comparison in the random geometric model. Again, Pearson and Spearman correlation coefficients of the degree sequence and betweenness centrality in the two scaffolds are compared. Each box is computed over a sample of 30 random geometric graphs, with parameters as reported on the x-axis: the pair (N,t) indicates a graph on N nodes sampled uniformly at random in the $[0,1]^2$ square. t is the connectivity distance threshold. The cyan x's and the green diamonds represent the average correlation value against the loose and minimal null models, respectively. The darker boxes in panels (a) and (b) report, for their respective model and for each metric and parameter values, the fraction of the sampled instances for which the Kolmogorov-Smirnov test was inconclusive (p value > 0.05). (c) Correlation tests for several network metrics show significant capabilities of the standard scaffold to reproduce certain statistical properties of the minimal one in C. Elegans. At the same time, due to different construction mechanisms, others are unreliable. (d) Scatterplot of the degree sequence of neurons of C. Elegans in the minimal scaffold versus in the loose one.

a sample of 30 random graphs of the reported model, with parameters as indicated on the x-axis.
For comparison, two null models are built for each instance of the minimal and loose scaffolds in the sample, by constructing an Erdős-Rényi random graph on the same vertex set, one with the same number of edges as the minimal scaffold, and one with the same number as the loose one. The correlation is computed of each statistic between the minimal scaffold and the loose null model and between the loose scaffold and the minimal null model. The average of these correlations is reported on the boxplots to act as a baseline value, highlighting that the two scaffolding procedures agree with each other by more than just statistical noise.
For a finer analysis, we have performed a two-sample Kolmogorov-Smirnov test comparing the distribution of the given metrics in the minimal and loose scaffolds, for all parameter values of the two random models. We consider the Kolmogorov-Smirnov test to be inconclusive if its p value exceeds a threshold of 0.05, in which case one cannot confidently reject the null hypothesis that the samples are drawn from the same distribution. In Fig. 3.8 panels (a) and (b), the darker boxes report for each parameter choice and metric the fraction of samples for which the test was inconclusive: in all cases except one, the KS test could not distinguish between the distribution of the graph statistic between the minimal and loose scaffolds, strengthening the indication of a good agreement between the two.

nPSO Random Graph Model

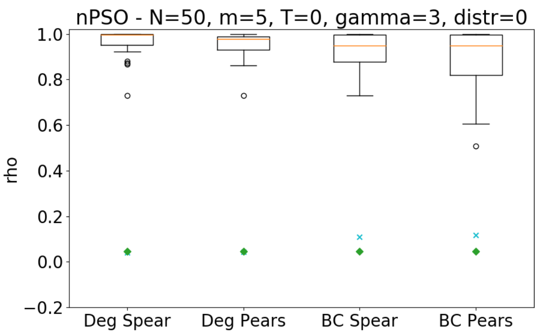
A modern random graph model, which has recently gained traction in network science for its ability to concurrently tune several parameters of interest in modeling real networks, is the Nonuniform Popularity-Similarity model. Introduced in [94], it builds upon a sequence of increasingly refined generative models to provide all the key structural properties of real-world graphs, such as scale-freeness, small-worldness and community structure. We therefore set out to employ it as benchmark in our comparison of the minimal and loose scaffolds.
In general, networks which display hyperbolic geometries tend to have a rather tree-like structure, with a certain scarcity of cycles. It is straightforward that, in the absence of a significant structure of persistent homology, the loose and minimal scaffolds will agree to high degree for at least two reasons: the low number of cycles forces the loose scaffold to localize onto the few available holes, hence resembling

the minimal, and secondly the scarcity of homology makes for a comparison between two mostly empty sets.

Following the lead of [95], we tuned the nPSO model parameters in order to empirically maximize the persistent homology structure, so as to make the comparison the most significant possible. As reported in Fig. 3.9, we observe again good ability of the scaffolds to proxy each other across the metrics analyzed, significantly higher than with respect to a null model, for a sample with parameters $N = 50, m = 2, T = 5, \gamma = 3$ and uniform distribution. A Kolmogorov-Smirnov test was also performed, as in the previous section, where a p-value higher that 0.05 indicates that the distribution of degrees and betweeness centralities in the minimal and loose scaffold cannot be confidently distinguished. This was the case for all the samples we tested.

nPSO Model Comparison

(a)



(b)

KS	Degree	BC
Fraction	30/30	30/30

(c)

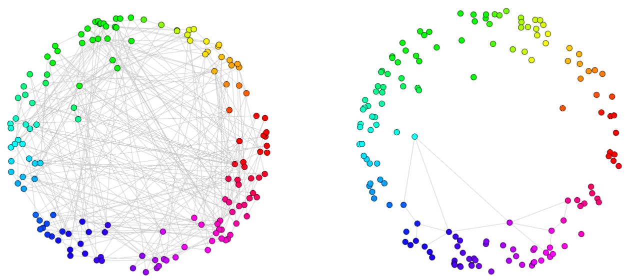


Fig. 3.9 Comparison of the minimal and loose scaffold for nPSO random model. (a) Degree sequence and betweenness centrality in the two scaffolds are compared using Pearson and Spearman correlation coefficients. Each box is computed over a sample of 30 nPSO instances, with the following parameters: 50 nodes, average degree 10 ($m = 5$), 0 temperature, power-law exponent $\gamma = 3$, and uniform distribution of angular coordinates. The cyan crosses and green diamonds represent the average correlations against the loose and minimal null models respectively, as in Fig. 3.8. In panel (b), the table reports, for the degree and betweenness centrality distributions, the fraction of Kolmogorov-Smirnov test that could not reject the hypothesis of the two samples coming from the same distribution. This has always been the case for each sampled instance and both metrics. (c) A graphical depiction of an instance of the nPSO model with parameters $N = 150, m = 2, T = 5, \gamma = 3$ and uniform distribution on the left. On the right, the corresponding minimal scaffold.

Conclusions

We provided a new method of network analysis and skeletonization, based on the computation of minimal homology bases. This new construction fills a significant gap in previous literature, in that it yields, in all but some *rare* pathological cases, a well-defined and unique subgraph, acting as a reasonable ground truth for comparison with the previous construction. It can be employed in a range of applications, both to identify crucial and weak links in a network, and to obtain compressed and topologically sound representations of the input. It also allows to evaluate the reliability of other scaffolding procedures with respect to said ground truth: we have observed that, for some applications, the loose scaffold can be deemed a sufficiently accurate tool, while not incurring in as cumbersome a computational load.
We foresee that the subject of homological skeletonization is not yet concluded. As mentioned, other approaches to finding canonical generators of homology are possible, One question that could spark from the present approach is the following: could one construct a sensible "entropy" functional on the space of cycles, so as to obtain by probabilistic methods a strictly unique, minimally-represented basis?

3.3 Homologically Persistent Skeleton

In the previous section we discussed extensively how one can apply the notion of minimal bases of homology towards the applications. In particular, we came to the conclusion that a skeletonization approach such as the minimal scaffold provides arguably the most refined information on the localization of (persistent) 1-cycles, but at the same time incurs in an often intractable computational cost.
Maintaining the notion of geometric minimality as a goal, we can approach the problem with different tools that compromise between accuracy in terms of homology localization and cost. In this setting we studied the work of Kurlin and various collaborators on the topic of *homologically-persistent skeletons*, shortened in *HoPeS*. These constructions were first developed for the 1-dimensional case and later generalized to arbitrary dimension d, and leverage the notion of minimum spanning tree, in particular the crucial properties of trees with respect to cycles. They sparked a series of papers on the topic, stemming from the work contained in [66].
We have provided a Python implementation of the 1-dimensional version of *HoPeS*. Further, in order to make this construction more conceptually similar to that of the minimal scaffold, we have proposed a modified version of HoPeS that more closely mimics the results of skeletonization via a homology basis, which we called *pruned* HoPeS.
Next comes a section where we study the behaviour of HoPeS as defined in the mentioned paper, when the point cloud under study is highly symmetrical. We show that some care must be taken when computing persistent homology in that circumstance.
Finally, we have applied this method in a similar fashion to the previous section, sketching a statistical comparison between the pruned HoPeS and the minimal scaffold. This part is work that I conducted jointly with Nicola Porru, a master student whose work I co-superivised and whom I thank.

HoPeS

The Homologically-Persistent Skeleton was first introduced in [66]. The construction is sketched as follows. Let C be a finite subset of a metric space. In the following we will consider $C \subset \mathbb{R}^n$, although the method is not limited to this circumstance. For example, we could equally work with a metric graph, that is a non-negatively

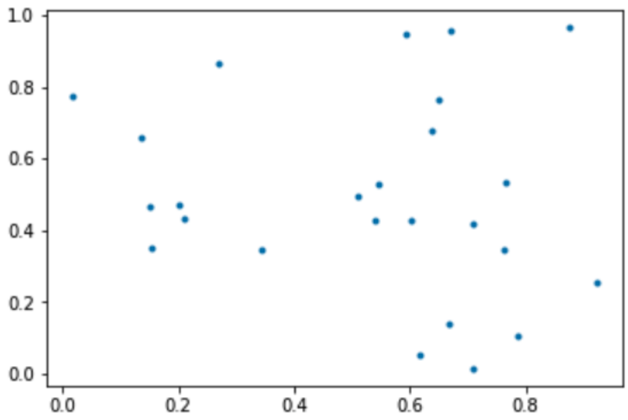


Fig. 3.10 A point cloud $C \subset \mathbb{R}^2$

weighted graph as in the previous section, with the additional requirement that weight obey the triangular inequality.
Let d_{ij} be the distance matrix of cloud C. Let $\mathcal{F}_0$ be the usual filtration of flag complexes induced by d_{ij} as in section (3.1.3), that is a Vietoris-Rips filtration built by considering the graph induced by $(d_{i,j})_{i,j}$ and, for each ε, computing the flag complex of the subgraph obtained by thresholding out edges of length higher than ε. Again inclusion maps are the obvious ones. In Figure 3.10 we can see an example point cloud in $\mathbb{R}^2$.

Finally, let MST(C) be a *minimum spanning tree* of the point cloud C.

Definition 56. (Minimum spanning tree) Let W be a weighted metric graph (that is, a graph whose weights form a metric). We call *minimum spanning tree* a subgraph $MST(W)$ of W that contains all vertices of W and such that

- $MST(W)$ is connected
- For any other such subgraph, the total length of its edges is $\geq$ than the total length of the edges of $MST(W)$.

Notice that by definition a tree is connected and contains no cycle. Then the minimum spanning tree is guaranteed to be a tree, as is has to be connected by definition, and the presence of any cycle would violate the requirement of minimality (since the graph is metric and therefore contains no zero-length edges). Since we assume to work with a point cloud $C \subset \mathbb{R}^n$, and this naturally induces a metric structure on C, we will equivalently write $MST(C)$ to mean the minimum spanning tree of

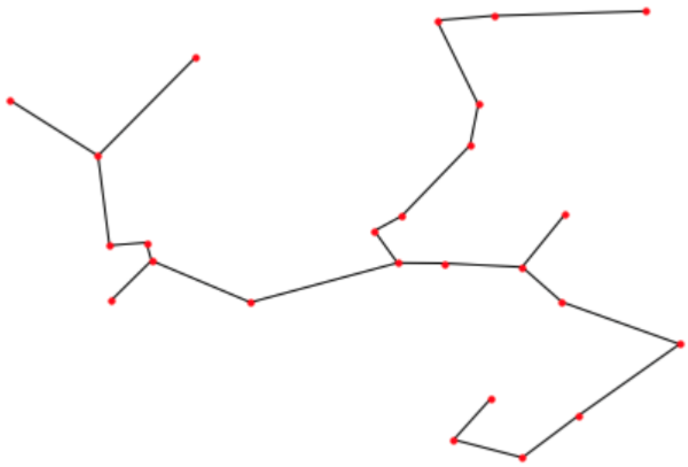


Fig. 3.11 The minimum spanning tree of point cloud C.

the corresponding metric graph. In Figure 3.11 we can see $MST(C)$, the minimum spanning tree of the point cloud in the example above.

We say a filtration is *simplex-wise* if it defines an injective map from the critical parameter values $\{\varepsilon_k\}_{k=1}^n$ into the simplices. In other words, if each step of the filtration amounts to adding exactly one simplex, therefore defining a strict total order on these. Notice that since the filtration is built from data, it is not restrictive to assume the persistence module is tame and the critical parameter values are a finite set.
The classical algorithm to compute persistent homology proceeds simplex by simplex, and therefore requires a strict total order on the simplices. In general filtrations computed from data need not be simplex-wise, as several simplices may be introduced at the same scale. One such filtration, as is $\mathcal{F}_0$, must be *refined* into a simplex-wise one, which we denote here by $\mathcal{F}_{SW}$, and assume that $\mathcal{F}_{SW}$ is a diagram of type $([n], \leq)$, regardless of the type of $\mathcal{F}_0$. We write that

$$\mathcal{F}_0 \preceq \mathcal{F}_{SW}$$

and say that $\mathcal{F}_0$ is refined by $\mathcal{F}_{SW}$, to mean that every complex $\mathcal{F}_0(\varepsilon)$ appears as some $\mathcal{F}_{SW}(i)$, but not necessarily the other way around.
In order to execute the persistent homology algorithm, the choice of this refinement

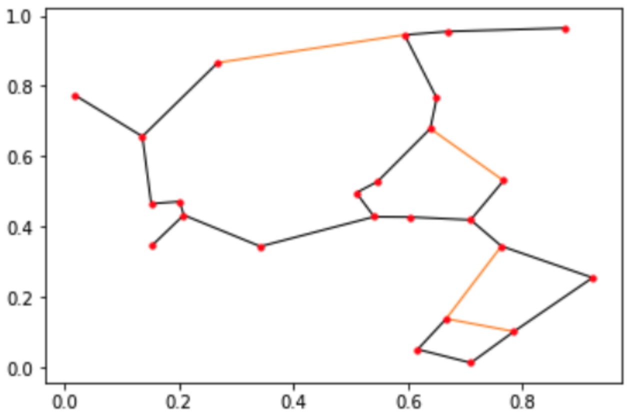


Fig. 3.12 The homologically persistent skeleton (HoPeS) of the point cloud C. Critical edges are highlighted in orange.

is irrelevant. However, we will see that for HoPeS, we must pay attention to the choice of refinement if several edges in C are the same length, to avoid pathological behaviour.

The reason we want a simplex-wise filtration is so that we can invert the function from parameter values to simplices into a function from simplices to parameter values. A Mayer-Vietoris-type argument guarantees that for a simplex-wise filtration, every step corresponds to either creating homology in dimension k or destroying homology in dimension $k-1$, and as the algorithm determines the effect of each simplex in creating or destroying homology, this association is necessary to obtain persistent pairs. When pairs are converted from being indexed over $\mathcal{F}_{SW}$ to being indexed over the original filtration $\mathcal{F}_0$, the arbitrary choice of a refinement comes undone and we revert to the appropriate values.

This being said, denote by $E^*(C)$ the set of edges in the point cloud C whose introduction generates a persistent homology class of dimension 1. As per [66], these are called *critical edges*, and it is a well-known fact that a minimum spanning tree of C contains none of them. To each critical edge, one can associate the persistence interval of the class of PH_1 that it generates.

Definition 57. (HoPeS, [66]) HoPeS(C) is defined as $MST(C) \cup E^*(C)$. Each element of E^* is decorated with its persistence interval.

In Figure 3.12 we see the HoPeS of C as in the previous examples. Critical edges are highlighted in orange.

Remark 1. The theoretical computational complexity of the HoPeS algorithm is dominated by the persistence algorithm, which has a theoretical running time of $O(n^3)$ in the number of simplices, whereas computing a minimum spanning tree costs no more than $O(n \log m)$ where n and m are the number of edges and vertices respectively. It is known that in practice the persistence algorithm often runs in slightly more than linear time, leaving us with a conservative estimate of a complexity that is less than quadratic, vastly more efficient than the minimal scaffold.

By definition, HoPeS consists of the (disjoint) union of a tree and a set of edges, one for each persistent homology class that exists throughout the filtration. By construction, it is clear that HoPeS is always a connected graph, and in particular we notice that it contains a set of cycles, connected by linear bridges, and with a branch-like structure expanding towards each node that does not participate into cycles. In other words, HoPeS has an arborescent structure mixing cyclic components with tree-like components.
If we think of the skeletonization approaches as the ones in the previous section, instead, these were built based only on the cyclic structure given by homology generators. Indeed, the scaffolds need not be connected graphs in general, as nodes that never participate into representative cycles of homology are in fact disconnected from the rest of the graph.
If we are keen on utilizing HoPeS as a less costly alternative to the minimal scaffold, the tree-like arborescent structure is actually superfluous, and this is the reason behind the modified definition of the *pruned HoPeS* that we give in the following.
Let C, $\mathcal{F}_0$ as before. Consider the following procedure:

Algorithm 3.1: Pruned HoPeS

Input: point cloud C ;
Result: Pruned HoPeS of C

PrunedHoPeS(C) := the empty graph on C ;
Compute $E^*(C)$ and $MST(C)$ as above ;
for $e \in E^*(C)$ **do**
    Add edge e to the MST of C: $C_e := MST(C) \cup e$;
    while C_e contains a vertex v of degree 1 **do**
        Remove from C_e the only edge coming into vertex v ;
    end
    PrunedHopes(C) := PrunedHoPeS$(C) \cup C_e$;
end
return *PrunedHoPeS*(C)

It works as follows: firstly, one computes the minimum spanning tree $MST(C)$ of the point cloud and the critical edges $E^*(C)$. Then one adds to the $MST(C)$ the critical edges *one at a time*, and at each time *prunes* the resulting structure. Adding one critical edge results in a graph that has exactly one cycle. The pruning is performed by identifying *leaves*, i.e. vertices with degree 1. Each time a leaf is found, the edge coming into it is deleted, and the operation is repeated until one is left with only the cycle. One saves this cycle, moves to adding to the $MST(C)$ the following critical edge, and prunes again, finding the next cycle. In the end, the union of these cycles is the PrunedHoPeS.

Definition 58. (Prune HoPeS) Given C as above, the output of algorithm 3.1 is the PrunedHoPeS of C.

In Figure 3.13 we can see the PrunedHoPeS of point cloud C as in the previous examples, with critical edges highlighted in orange. It more closely resembles the skeletonization approaches based on generator cycles of persistent homology; for example, we see it is disconnected.

Consistency of refinements

As we mentioned above, running the persistent homology algorithm normally requires refining a filtration to a simplex-wise one. For the purposes of computing persistence pairs, how one breaks the ties is essentially immaterial, because this

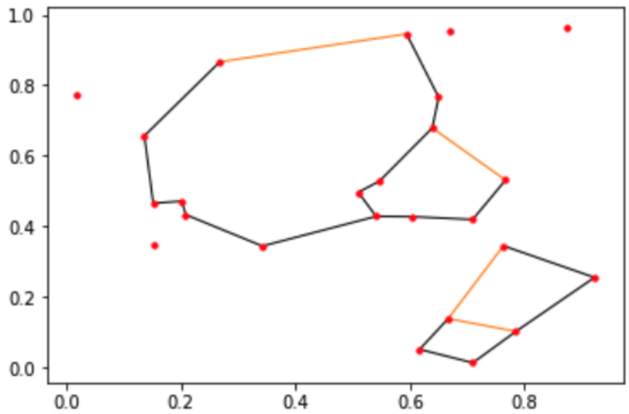


Fig. 3.13 Our proposed construction of a *pruned* homologically persistent skeleton.

operation is then reversed when returning the result.
Here, instead, we are interested in identifying the critical edges, and for the whole algorithm to be correct we must be sure that these edges are *not* contained in the minimum spanning tree. If we assume that all edges in the point cloud are of different length, then the filtration is automatically simplex-wise and we need not worry about this fact. If instead we have a point cloud that is highly symmetrical, this could cause a problem due to the MST not being unique.

Example. Consider for example the set of four points arranged in a square. Four MST's are possible, corresponding to each of the four perimeter edges that can be left out. At the same time, since the four edges enter into $\mathcal{F}_0$ at the same time, we have many possible refinements of $\mathcal{F}_0$ into $\mathcal{F}_{SW}$, and the choice of which edge is taken as the critical one depends on which is introduced *last* into $\mathcal{F}_{SW}$.
If the choice of the refinement $\mathcal{F}_{SW}$ is made randomly and independently of the construction of the MST, one has a three out of four chance to end up in a situation where the last edge to enter $\mathcal{F}_{SW}$ is contained into MST. The result is catastrophic, because the resulting HoPeS contains no cycle.

The choice of a MST for C can be used to construct a refinement $\mathcal{F}_{MST} \succeq \mathcal{F}_0$, where edges that belong to the MST appear *before* edges of the same weight that do not. When the choice of an MST is not unique because more than one exists, this refinement is strict.

Lemma 2. *If $\mathcal{F}_0 \preceq \mathcal{F}_{MST} \preceq \mathcal{F}_{SW}$, then no critical edge for $\mathcal{F}_{SW}$ belongs to the MST.*

Proof. Suppose otherwise, so there exists $e \in MST$ that is critical for $\mathcal{F}_{SW}$. We must show that $\mathcal{F}_{MST} \not\preceq \mathcal{F}_{SW}$. Let $\mu(e)$ be the length of edge e, and consider $\mathcal{F}_0(\mu(e))$, the complex containing all simplices of diameter $\leq \mu(e)$. By the construction of $\mathcal{F}_{MST}$, there exists an integer $i \in \mathbb{N}$ such that $\mathcal{F}_{MST}(i)$ is a sub-complex of $\mathcal{F}_0(\mu(e))$ containing all simplices of diameter $< \mu(e)$, and only those edges of length $\mu(e)$ that belong to the MST. It will suffice to show that there is no $j \in \mathbb{N}$ such that $\mathcal{F}_{MST}(i) = \mathcal{F}_{SW}(j)$.
First, notice that by construction $\mathcal{F}_{MST}(i)$ must contain edge e. Then, notice that since e is critical for $\mathcal{F}_{SW}$, if $\mathcal{F}_{SW}(j)$ contains e is must contain a cycle (in the sense of a 1-chain) through e with birth time $\mu(e)$. So if it were that $\mathcal{F}_{MST}(i) = \mathcal{F}_{SW}(j)$, then $\mathcal{F}_{MST}(i)$ must contain a cycle through e. In other words, e does not kill a connected component in $\mathcal{F}_{MST}(i) = \mathcal{F}_{SW}(j)$. But then $MST \setminus e$ is another spanning tree with length strictly less than MST, and this is a contradiction. $\square$

If instead $\mathcal{F}_0 \preceq \mathcal{F}_{MST}$ and $\mathcal{F}_0 \preceq \mathcal{F}_{SW}$ but $\mathcal{F}_{MST} \not\preceq \mathcal{F}_{SW}$, i.e. the simplex-wise filtration implicitly built by the persistence algorithm is not informed by the choice of tree- and non-tree-edges, then pathological configurations such as the one described in the example may appear.

To ensure that $\mathcal{F}_{MST} \preceq \mathcal{F}_{SW}$, one must first produce any MST and then reorder $\mathcal{F}_{SW}$ so that within any weight class (i.e. fibre of the map that assigns a weight to every simplex), first come the edges that belong to the MST, then come the non-tree edges, and finally all triangles. One such list of simplices induces a simplex-wise filtration that obeys the required constraint $\mathcal{F}_{MST} \preceq \mathcal{F}_{SW}$, and therefore produces a correct output.

Comparison with the minimal scaffold

We finally ran the pruned HoPeS model to compare it against the minimal scaffold presented above. We must firstly observe that while the minimal scaffold only requires a semi-metric space to be defined, [66] requires that the space on which to compute the persistence skeleton be metric. Therefore, many of the data sets or graph models presented above cannot be applied. The Human Connectome and the C.Elegans data sets are correlation networks, and as such do not normally form metric spaces (although they could be transformed into one). The Watts-Strogatz

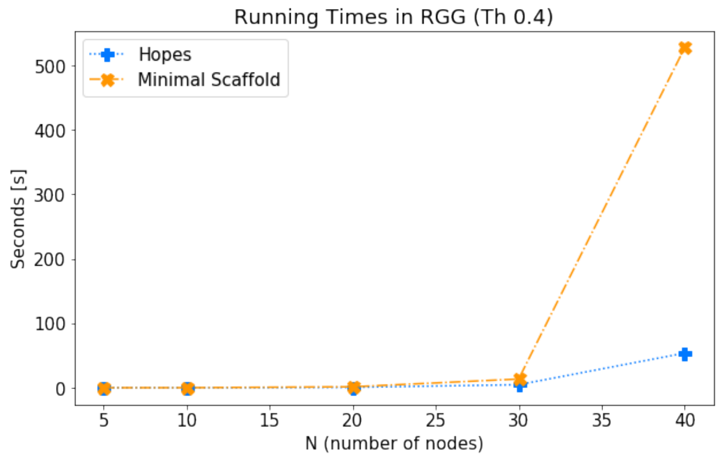


Fig. 3.14 Comparison of running times between pruned HoPeS and the Minimal Scaffold over Random Geometric Graphs on N nodes, with a threshold value of 0.4.

and nPSO models also do not enjoy a metric structure. Therefore, we limit our comparisons to the Random Geometric Graphs.

As a first step, we compare the computational complexity in terms of computation time between pruned HoPeS and the minimal scaffold, on a family of Geometric Random Graphs on N nodes, with a fixed threshold of 0.4. We can see the result in Figure 3.14, where it is apparent that while for very small graphs the computational overhead dominates the difference between the two models, as soon as the number of nodes (hence edges) increases slightly the minimal scaffold sharply becomes much more difficult to compute.

Then, we have performed an exploratory analysis of the correlation between certain graph statistics of the minimal scaffold and of the pruned HoPeS, in the same spirit as done in Section 3.1. We present results in Figure 3.15, where we can see the boxplots of the correlation values ρ over a family of 30 geometric random graphs whose parameters are reported on the x-axis. The first panel reports results for the correlation of the degree sequences, and the second for betweenness centralities.
We report a generally positive correlation between the two models, although the magnitude of this correlation varies widely from instance to instance, thereby suggesting that further and larger experiments are needed to cope with the variance. Additionally, it would be beneficial to perform a significance test onto the correla-

tions obtained.

We foresee to explore these avenues in subsequent work.

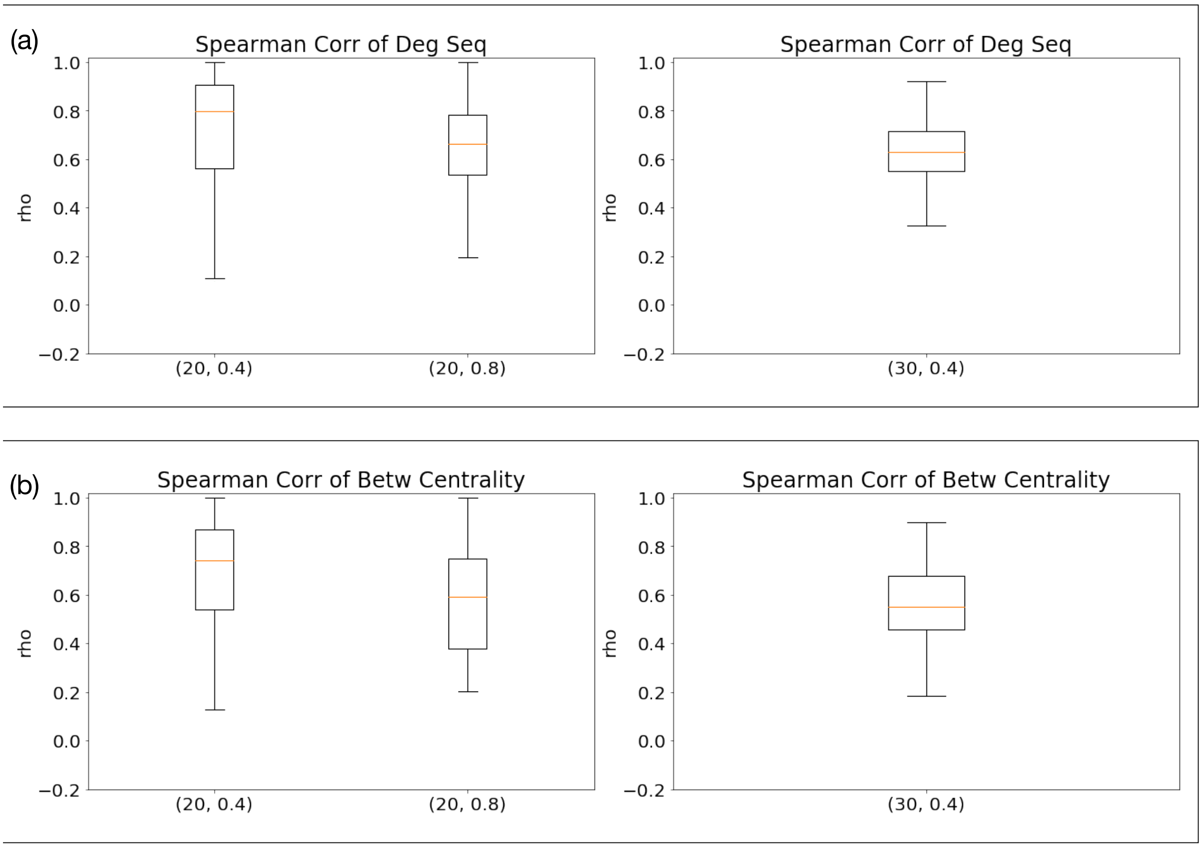


Fig. 3.15 (a) Boxplot of the correlation coefficients of the degree sequences of the pruned HoPeS and of the minimal scaffold computed on a family of 30 RGG instances, with parameters reported on the x-axis. For each boxplot. the first parameter on the x-axis reports the number of nodes in the graph, and the second reports the threshold value. (b) Boxplot of the correlation coefficients of the betweenness centrality of the pruned HoPeS and of the minimal scaffold computed on a family of 30 RGG instances, with parameters reported on the x-axis. For each boxplot. the first parameter on the x-axis reports the number of nodes in the graph, and the second reports the threshold value.

3.4 Canonical representatives via Alexander duality

In this section we present some material pertaining a research line that is ongoing work between myself, Alessandro De Gregorio and Sara Scaramuccia. As a whole, it is at a preliminary level and mostly contains a review of the literature about the topic of searching for *canonical* representatives of homology cycles by leveraging the correspondence between zero-homology and $d-1$ homology given by the famous theorem by Alexander. Alongside those, we present a few ideas, results and conjectures of our own.
The main point is an algorithm to compute tight homology representatives under the hypotheses of Alexander duality. The idea was already present in the PhD thesis [96] by Benjamin Schweinhardt, and has also been touched upon in [71].

Related Works

Several works have historically touched on the topic of homology localization ([97–100]).
Here, we mention in particular the PhD thesis [96] by Benjamin Schweinhardt. It already contains the idea of having a canonical representative by relating codimension 1 object to connected components in the complement via Alexander duality. Then, it leverages the properties of zero-dimensional homology, namely the ability to identify a 0-homology class with a set of points, to define a canonical representative. In doing so, however, it assumes the knowledge on a triangulation of the sphere and of a subcomplex of that triangulation. Our approach, on the other hand, works without the assumption of having a triangulation of the sphere of which our data is a subcomplex. In other words, it saves the need to *extend* a given simplicial complex to a triangulation of a sphere.
In the work [71] by Obayashi, the author has the same goal as [96], but tries to extend it to the case of codimension > 1, for which duality no longer holds. It works

by relaxing the "exact" problem

$$\begin{aligned} \text{minimize } |z|_0 \text{ s.t.} \\ z = z_i + \partial \omega \\ \omega \in C_2(K, \mathbb{Z}_2) \end{aligned}$$

into the linear problem (l^1):

$$\begin{aligned} \text{minimize } |z|_1 \text{ s.t.} \\ z = z_1 + \partial \omega \\ \omega \in C_2(K, \mathbb{R}) \end{aligned}$$

It is heuristically known that this should approximate the solution, but it is not an exact algorithm.

Alexander duality

Let $H_k(\cdot, \mathbb{F})$ (respectively $H^k(\cdot, \mathbb{F})$) be the k-homology functor (respetively the k-cohomology functor) with coefficients in a field $\mathbb{F}$. Consider the d-dimensional sphere S^d.
It is famous that

Theorem 4. (Alexander duality, [101])
Let X be a proper, non empty subset of S^d, and assume that (S^d, X) is triangulable. Then there is an isomorphism

$$\tilde{H}^k(X, \mathbb{F}) \cong \tilde{H}_{d-k-1}(S^d \setminus X, \mathbb{F})$$

The theorem states an isomorphism between the reduced cohomology in degree k of X and the reduced homology in degree $d-k-1$ of the complement of X in the sphere S^d. As mentioned above, we are especially interested in the case $k = 0$, as zero (co)homology enjoys special properties in terms of class representatives, and this yields the isomorphism

$$\tilde{H}^0(X, \mathbb{F}) \cong \tilde{H}_{d-1}(S^d \setminus X, \mathbb{F})$$

We see, therefore, that homological features in dimension $d-1$ (for a subset of the d-sphere, which we often call *codimension* 1 features) are also amenable to especially nice properties. Specifically, 0-dimensional homology is particularly convenient, in that it has canonical representatives for each class (i.e. component). This is due to the fact that cycles in dimension zero are 1-to-1 with vertices as $\partial_0 = 0$, so $Z_0 := \ker \partial_0 = C_0$. This entails that a basis of H_0 is a *subset* of the standard basis of C_0.

In the following, the requirement that the spaces be triangulable will be thoroughly enforced. In fact, as we are mostly interested in applying the method to data, we will only deal with *finite* simplicial complexes, without ever considering the singular or homology manifolds for which the theorem is actually valid.
As such, let L denote a d-dimensional, finite simplicial complex that is a triangulation of the d-sphere, i.e such that we have a homeomorphism

$$L \cong S^d$$

Recall we write $\trianglelefteq$ to denote a simplicial subcomplex. Let $K \trianglelefteq L$, and call $\overline{L \setminus K}$ the closure (in L) of $L \setminus K$, i.e. the smallest subcomplex of L that contains $L \setminus K$. Alexander duality specializes into

$$\tilde{H}^0(K, \mathbb{F}) \cong \tilde{H}_{d-1}(\overline{L \setminus K}, \mathbb{F})$$

Now, since we consider reduced (co)homology with coefficients in $\mathbb{F}$, and since we assumed all simplicial complexes to be finite, by the universal coefficients theorem and the properties of finite-dimensional vector spaces the above translates into the isomorphism

$$\tilde{H}_0(K, \mathbb{F}) \cong \tilde{H}_{d-1}(\overline{L \setminus K}, \mathbb{F}) \tag{3.5}$$

between 0-dimensional reduced homology and $d-1$-dimensional reduced homology. This is the version that we will be interested in, as it allows us to leverage the canonical representatives for H_0, that is a *good* choice for a preimage of the homology map $[\cdot] \ : C_0 \to H_0$, to form a map

$$\tilde{H}_{d-1} \xrightarrow{\cong} \tilde{H}_0 \longrightarrow C_0$$

where elements of $\tilde{H}_0$ are connected components up to one "framing" component.

We will mostly consider the case $d = 2$. In fact, we will assume that the data we are given is in the form of a planar simplicial complex, in the following sense. Assume we are given a triangulation of a compact subset D of $\mathbb{R}^2$, i.e. a set homeomorphic to the 2-disk. For brevity, we will again call D the simplicial complex given by this triangulation. Consider the boundary of D, which is a 1-chain ∂D, introduce a virtual point at infinity x_∞ and consider the *cone* of ∂D with x_∞. This is homeomorphic to the 2-sphere S^2.

$$S^2 \cong D \cup \partial D * x_\infty$$

where $*$ denotes the coning operation. The cone of a point x_∞ with a simplicial complex K is defined as the set of simplices $[x_\infty \sigma_0 \dots \sigma_p]$ for every $[\sigma_0 \dots \sigma_p]$ that is a p-simplex in K, plus all their faces ([101]).
So normally our data is of the form $L := D \cup \partial D * x_\infty$, and we will consider a subcomplex $K \trianglelefteq L$. Moreover, we will assume that the subcomplex K does not contain x_∞, i.e. it is a *planar* subcomplex of D.
This whole setting allows us to talk about the homology classes in dimension 1 of K in terms of the connected components (up to the framing component) of $\overline{L \setminus K}$.

Canonical representatives

Assume $K \trianglelefteq L$ as above are given. Consider a class $[c] \in H_{d-1}(K)$, whose representative is any cycle $c \in Z_{d-1}$. The power of Alexander duality lies in the fact that it allows us to canonically identify a set of vertices in the complement. Namely, call Al the isomorphism in Alexander duality. Then $[c]$ is mapped to a connected component γ of $\overline{L \setminus K}$:

$$[c] \xmapsto{Al} \gamma \in \tilde{H}_0(\overline{L \setminus K})$$

For 0-dimensional homology, there is a natural way to define a correspondence between 0-chains and vertices. Call this $\mathrm{supp}\,(:) C_0 \to K^0$. Now, γ is *uniquely* represented by the subset of *all* vertices of $\overline{L \setminus K}$ whose chain is mapped to γ by the quotient map H_0. Denote this subset by vC. In formulae,

$$vC := \mathrm{supp}\left(H_0^{-1}(Al([c]))\right)$$

where by H_0^{-1} we mean the preimage of the homology map.
Now, call sC the full subcomplex of $\overline{L \setminus K}$ induced by the vertices belonging to vC. That is, a p-simplex $\sigma = [v_0, \ldots, v_p]$ belongs to sC iff all its $v_0, \ldots, v_p$ belong to vC.
Call $\tau_{[c]}$ the d-chain

$$\tau_{[c]} := \sum_{\substack{\sigma \in sC \\ \dim \sigma = d}} \sigma \tag{3.6}$$

This is, the sum of chains of all top-dimensional simplices contained in the full subcomplex corresponding to a connected component. We are now ready to define the notion of canonical cycle.

Definition 59. (Canonical cycle)
The $(d-1)$-cycle $\partial_d \tau_{[c]}$ is *the* canonical cycle of class $[c] \in H_{d-1}(K)$.

Uniqueness of the canonical cycle is automatic by the uniqueness of τ, which is guaranteed by duality and by the properties of H_0. We remark that formally the canonical cycle is an element of $C_{d-1}(\overline{L \setminus K})$. Since in the above we postulated explicit knowledge of the inclusion $K \hookrightarrow L$, it is immediate to construe the canonical cycle as an element of the desired space $C_{d-1}(K)$.
At the same time, this represents a limitation to concretely using this approach in practice. If one is only given data in the form of a d-dimensional triangulated manifold K, to construct the canonical cycle in this way one must first *extend* K to L, that is construct a compatible triangulation of S^d from K.

Tight representatives

Now assume we are given a complex K that is a triangulation of a subset of a compact D as above, i.e. $D \cup \partial D * x_\infty \cong S^d$. In particular, L is a d-manifold. We must define a notion of simplices being "inside" or "outside" of a cycle. The precise definition of a point being inside or outside of a cycle is the key problem that we wish to address in order to make the following into a precise definition. For now, let us assume that we have an agreed notion of inside and outside, or equally, of containment. We will say more about it in the following, but its solution is ongoing work.
Let us define

Definition 60 (tight cycle)**.** A $(d-1)$-cycle c in K as above is called *tight* if no d-simplex of K is "inside" of c.

Definition 61 (tight cycle with geometric realization)**.** A $(d-1)$-cycle in K as above is called *tight* if it does not contain any point in the interior of a d-simplex of K.

We have given two definitions of a tight cycle, one leveraging the geometric containment, and one the abstract notion of inside. We remark that in both cases it is not trivial to give a precise meaning the these terms. For a cycle that is a simple plane curve the distinction is clear, but cycles that have self intersections are more complicated.
One approach that we are investigating is as follows: a simplicial cycle decomposes into a number of polygons. If the cycle is self-intersecting, this number is greater than 1. A method called ray-tracing allows to define a suitable concept of *inside*: given a point, consider a half-infinite ray centered at the point. For each polygon, if the ray intersects the polygon an even number of times then the point is called *outside* the polygon, otherwise it is called *inside*. Finally, the point is *inside* the cycle if it is inside an odd number of its polygons, *outside* otherwise.
One problem with this approach is that a decomposition into simple polygons is not unique, and it is not even always formed by the same number of polygons. Then one first hurdle is to ensure that this definition is consistent independently of the choice of polygonal decomposition.

Tightening is in fact an operation on cycles. Indeed, given any cycle c, one can easily obtain a tight representative, specifically

$$t_c := c + \sum_{\substack{\dim\sigma = d \\ \sigma \text{ "inside" } c}} \partial_d \sigma \tag{3.7}$$

Proposition 1. *t_c is tight and homologous to c*

Proof. The two cycles differ only by boundaries. □

Proposition 2. *The tight representative is unique*

Proof. If c_1 and c_2 are homologous cycles, so that

$$c_1 = c_2 + \sum_{\sigma \in I} \partial\sigma$$

then for each σ in I exactly one of the following claims is true:

- σ is inside c_1 and outside c_2;
- σ is inside c_2 and outside c_1.

Let us denote with $\#(c,\sigma)$ the number of polygons of c that contain σ. Consider c_2 and $c_2+\partial\sigma$. Suppose that $\partial\sigma$ and c_2 have no edge in common. If σ is inside c_2, then $\#(c_2,\sigma)$ is odd. When we add $\partial\sigma$ to c_2, we have that $\#(c_2+\partial\sigma,\sigma) = \#(c_2,\sigma)+1$. □

Theorem 5. *Each homology class h of $H_1(K)$ has a unique tight representative.*

Proof. Suppose to have two homologous tight cycles, $t_1 \neq t_2$. Since they are homologous, it must be $t_1 = t_2 + \sum_{\sigma\in J}\partial\sigma$. But then by the previous claim, if J is not empty there must be a σ inside t_1 and outside t_2 or vice versa, but this is impossible, since by construction, t_1 and t_2 do not contain any simplex inside themselves. Therefore, $t_1 = t_2$ and the tight representative is unique. □

Theorem 6. *Every canonical cycle as given by Alexander duality is tight (in K)*

Proof. Since the canonical cycle is in $C_{d-1}(L\setminus K)$, it contains no d-simplices of K □

We have sketched some concepts regarding canonical representatives via Alexander duality. Most of this material needs further analysis, and we foresee to work on it in the future.

Chapter 4

Interval Bases and persistence module decomposition

This chapter is based on work that I carried out jointly with my advisor Francesco Vaccarino and Alessandro De Gregorio and Sara Scaramuccia.
The focus in this chapter drifts away from the topic of homology representatives, that is choosing a good preimage for the homology quotient map, and towards the goal of choosing *good* (in a sense that will be discussed) generators *within the persistent module*.
A prototypical example of this problem appeared in the previous chapter, when discussing the uniqueness of the minimal scaffold, namely when considering three homology classes connected by a linear relation of the form $[A] = [B] + [C]$. The same situation appears a fortiori when considering persistence modules as opposed to "single" homology vector spaces, with the due modification that one does not consider linear relations over an $\mathbb{F}$-vector space, but rather over an $\mathbb{F}[x]$-module. This type of relation encodes homology classes
In that setting, the existence of such a relation proved troublesome to the task of defining a *unique* homological skeleton; however, it relates directly to the problem of choosing a criterion for picking basis for a vector space, a problem that is known to be fickle and can essentially only be tackled by enforcing some extrinsic criterion. This is even more true in the case of modules. The topic of choosing generators, or giving a *presentation*, is in itself a vast one. Considering that, in general, modules do not even admit a basis, the problem is larger than for the case of vector spaces. However, we will see that the structure maps in a persistence module do provide

criteria to operate such a choice, and this is indeed the goal of the work exposed here.

The central part this chapter deals with the decomposition of persistence modules. It necessarily stems from the fundamental theorem linking topological data analysis to representation theory, namely the theorem by Zomorodian and Carlsson establishing the equivalence of categories between one-parameter, equioriented, finitely-presented persistence modules over a ring R and $R[x]$ modules. This result lays the ground for the application of the Krull-Schmidt-Remak theorem, guaranteeing that a persistence module admits an essentially unique (up to isomorphism and reordering) structure of direct-sum *indecomposables*.

For the goal of computing persistence, obtaining such a decomposition is enough, as it allows to read off the persistence diagram, which is usually what one is interested in. Hence, the *classical algorithm* for persistent homology, for efficiency, performs a left-to-right, simplex by simplex reduction of the boundary matrix which pairs simplices identifying persistence pairs. Additionally, the reduced matrix contains generating cycles in the form of suitable k-chains whose boundary is zero, corresponding to homology classes which *generate* the persistence modules.

The output of this algorithm, however, does not take into account the structure of the *relations* between the computed generators. The chosen cycles, or rather their corresponding persistent homology classes, generate the whole persistence module through the action of the structure maps. However, the behaviour at their death points does not enjoy any particular property.

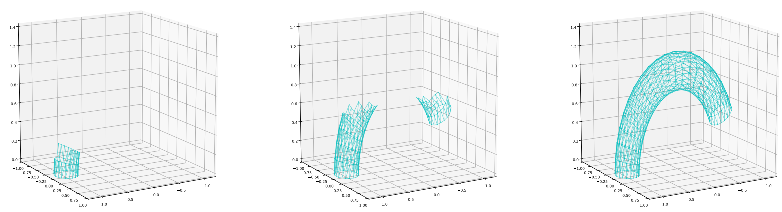
We have started from this consideration when searching for a different type of generator set, that we call an *interval basis.* These vectors not only generate the persistence module, but additionally enjoy injectivity properties along the structure maps, or in other words take into account the structure of the relations between them, so as to make them as simple as possible. That is, diagonal.

In the following, largely based on the preprint [102], we describe the concept of interval basis, and propose a parallel algorithm to compute one from a DAPM. We compare it to the previous constructions in the literature, namely studying its relationship with the primary decomposition of a module and with the graded Smith Normal Form. We then show that our step-wise parallel algorithm obtains an average computational advantage over the SNF version.

We conclude the study by showing a full parallel pipeline for the computation of persistent homology over the reals, by leveraging the Hodge decomposition and the isomorphism between harmonic representatives and homology. This allows to compute an interval basis simultaneously over each homogeneous component of the persistence module, starting from a filtered chain complex, replicating the traditional TDA pipeline.

4.1 Parallel decomposition of persistence modules through interval bases

Consider a (discrete, algebraic) persistence module $\mathcal{M} = \{M_i, \varphi_i : M_i \to M_{i+1}\}_i$ over a field $\mathbb{F}$ (DAPM), indexed over the natural numbers. In this section we will call each M_i the i^{th} *step* and each φ_i the i^{th} *structure map*. Assume, as per the background, that the module is tame or equivalently of finite type, i.e. that each $\mathbb{F}$-vector space M_i is finite dimensional and only a finite number of maps among the φ_i's are not isomorphisms.
We remark that the underlying structure is that of an equioriented quiver, in particular we do not consider zigzag modules in the following. By the fundamental representation theorem above, each $\{M_i, \varphi_i\}$ can be decomposed, up to ordering, into a direct sum of interval modules so that $\mathcal{M}$ is completely described by its *persistence diagram* or *barcode*. Furthermore, this decomposition is stable with respect to noise if the PM is made of the homology groups of a geometric sampling.
More explicitly, a persistence module can be written uniquely, up to reorderings of



(a) Step 0 (b) Step 1 (c) Step 2

Fig. 4.1 A 3-step filtration by sublevel sets, with respect to the the z coordinate, of a half torus.

summands, as a direct sum of persistence (sub)modules

$$\mathcal{M} \cong \bigoplus_m I_{[b_m,d_m]}, \tag{4.1}$$

where each term $I_{[b_m,d_m]}$, called *interval*, is an indecomposable persistence submodule depending on the indexes $b_m \leq d_m$. The multiset of pairs $\{(b_m,d_m)\}$ forms the barcode.
Under the isomorphism in Eq.(4.1), for each m, one might search for an element v_m in $\mathcal{M}$ which, through the application of the structure maps, generates the submodule $I(v_m)$ corresponding to $I_{[b_m,d_m]}$.

In this work, we call a set of generators $\{v_m\}_m$ of $\mathcal{M}$ an *interval basis* if it satisfies

$$\mathcal{M} = \bigoplus_m I(v_m) \cong \bigoplus_m I_{[b_m,d_m]}. \tag{4.2}$$

The classical algorithm to construct the barcode of a finite type persistence module ([8]) works at the level of the chain complexes and thus is limited to persistence modules induced by homology. The algorithm yields the pairs $\{(b_m,d_m)\}$, and can be used to extract a set of homology generators, as done in [103]. However, the extracted set of generators does not generally satisfy eq. (4.2).
For example, consider a filtration of the half torus such as the one depicted in fig. 4.1, which will be our running example throughout the paper. The classical algorithm correctly identifies the persistence pairs, but returns the generators depicted in fig. 4.2, in accordance with the so-called *elder rule*. These generators do not induce a decomposition such as the one in eq. (4.2), because their images in the final step of the filtration become homologous. Algebrically, this entails a linear relation between non-zero elements of $\mathcal{M}$ and thus the sum of interval modules in eq. (4.2) is not direct.
The principal contribution of this chapter is to provide a procedure to build an interval basis. From an interval basis, it is trivial to compute the persistence pairs. As an example, for the case of 1-homology of the half torus, a set of generators forming an interval basis is depicted in fig. 4.3.

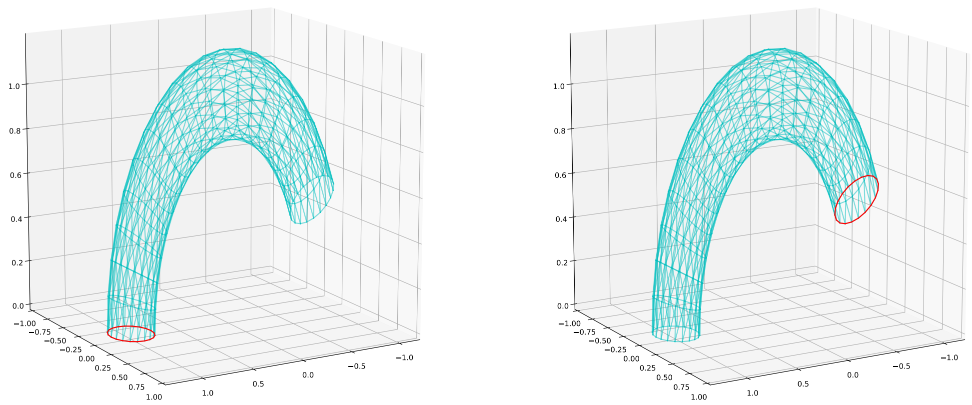
With the classical approach one is only allowed to compute the decomposition of a persistence module that is the homology of a filtration of simplicial complexes.

Moreover, the algorithm intrinsically works with objects (chain groups and chain maps) whose dimension is typically much larger than that of their homology module. Our proposed method, instead, addresses the case of finding a decomposition of a general, i.e. not necessarily a homology, persistence module. As such, we work directly at the level of $\{M_i, \varphi_i\}$. As a consequence, the bulk of our algorithm can be performed independently for each M_i, i.e. is largely parallelizable.

We can summarize the main contribution with the following result

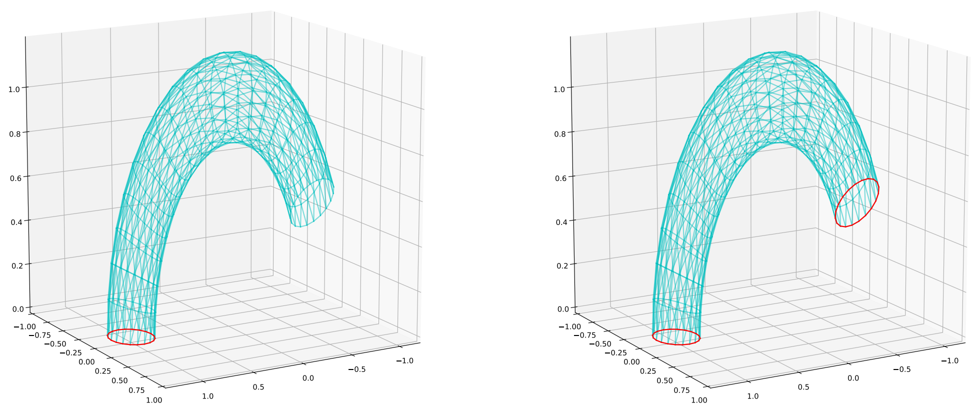
Theorem. *Given a finite-type $\mathbb{F}$-persistence module, the output of alg. 4.3 provides an interval basis.*

After giving a specific instance of our algorithm for the case of *k-persistent homology modules*, i.e., persistence modules obtained through the k-homology functor applied to a filtration, we specialize our procedure to the case where the chain spaces are Euclidean, i.e. have a scalar product. For example, we address the case of k-persistent homology modules with coefficients in $\mathbb{R}$, which can be built from the kernel of the k Hodge Laplacian operator with respect to a filtration of simplicial complexes. We construct the induced maps between kernels and, by the known properties of the Hodge Laplacian, prove it to be equivalent to the k-persistent homology module. We then apply our decomposition to obtain an interval basis, which is shown in fig. 4.4. This furthers the recent trend of exploring the interplay between topological data analysis and the properties of the Hodge Laplacian [104–106].



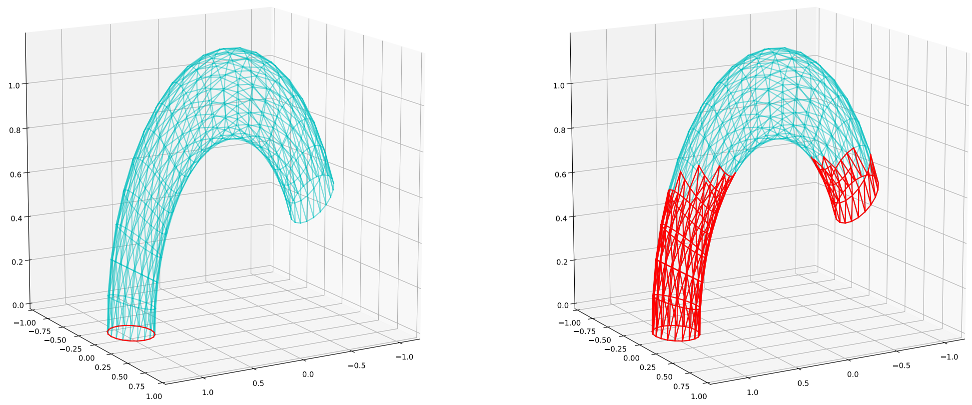
(a) Generator 1 (b) Generator 2

Fig. 4.2 Generators, classical algorithm.



(a) Generator 1 (b) Generator 2

Fig. 4.3 Interval basis generators for the *k*-persistent homology module.



(a) Generator 1 (b) Generator 2

Fig. 4.4 Interval basis generators for the harmonic persistence module

Contributions and related work

We sketch the original contents of this chapter and provide a framing within the literature on the subject:

1. An algorithm for the decomposition of any DAPM over any field. We do not assume the module is induced by homology. For example, the module might derive from any sequence of simplicial maps.
2. The generators we provide are such that a decomposition of the persistence module is given by the interval modules they each generate. We call this an *interval basis*.
3. Our algorithm is parallelizable for each step of the module.
4. A framing of the topic of interval bases within the literature on the primary decomposition of a module. Specifically, we expand on the existing literature, and provide computational guarantees.

We further provide a parallel algorithm to construct a persistence module from a general collection of simplicial maps. In the special case of real coefficients, we use the Hodge Laplacian as a construction method for homology groups and maps between them. We therefore specialize the decomposition algorithm.

The original approach to the decomposition of persistence modules arising from filtrations of simplicial complexes was introduced in [107], based on the simplex pairing approach of [108]. These algorithms were limited to working with $\mathbb{Z}_2$ coefficients and simplicial maps deriving from inclusions of simplicial complexes. Later, this framework was extended to arbitrary coefficient fields in [109], which established the notorious correspondence between persistence modules and graded modules over a PID. It is immediate to construct a set of generators from the output of these algorithms. However, they do not induce an interval basis decomposition of the module. A different approach to attempt a principled choice of homology generators is to introduce a notion of geometric minimality, as was done in [73, 67, 110].
The distributed computation of the persistence pairs has been addressed in [111], where arguments from discrete Morse theory are employed to observe that the reduction of a boundary matrix obtained from a filtration of simplicial complexes can be performed in *chunks*. The chunk algorithm can, like the one proposed here, be run in parallel for each filtration step, and is incremental in that it only considers cells that are born within the step itself. Its objective, however, is to compute the persistence pairs of a filtration, whereas our proposal works, for general (albeit monotone) simplicial maps, does not need to consider the (typically much larger) space of k-chains, and computes an interval basis.
Subsequently, the case of a monotone sequence of general simplicial maps was addressed in [112], which relies on the notion of simplex annotation. In this way, and by interpreting each simplicial map as a sequence of inclusions and vertex collapses, a consistent homology basis can be maintained efficiently. Later, in [113], a variation of the coning approach of [112] is proposed, which takes a so-called *simplicial tower* and converts it into a filtration, while preserving its barcode, with asymptotically small overhead. Our approach to the problem is different in that we do not need to work at the level of simplicial complexes or maps, and in fact our decomposition is independent from the persistence module being induced by a filtration of simplicial complexes.
To our knowledge, the first paper to deal with persistence module decompositions without assuming it arises from a family of inclusion maps was an instance of topological persistence whose underlying structure is not monotone: the so-called *zigzag persistence*([114]). Zigzag persistence is based on a sequence of structure maps

whose direction is not necessarily left to right. Its authors provide an incremental algorithm which computes a decomposition of the zigzag module, without focusing on its generators. Our proposal differs in that we are restricted to the case of monotone persistence, but in turn our focus is on the construction of an interval basis, and on a parallel algorithm.
An analogous point of view is taken in a recent paper by Carlsson et al. ([115]). The decomposition of the general zigzag module is tackled from the matrix factorization viewpoint. Starting from a type-A quiver representation, their goal is to compute a basis for each homogeneous component of a persistence module in such a way that the structure maps are represented as echelon form matrices. In other words, the matrix associated to the quiver representation is in canonical form. Our algorithm can decompose the module while avoiding the change of basis operations that are needed for the construction of the echelon form. Furthermore, by specializing to monotone persistence modules we can avoid the divide and conquer approach and instead perform computations in parallel across all steps in the persistence module.

A method to compute an interval basis can also be found in the literature via an apparently rather different path. As mentioned in the above, an interval basis corresponds to a special choice of generators in a minimal presentation of the $\mathbb{F}[x]$-module. In the corresponding section below, we piece together work by Corbet and Kerber [12], where one is shown how to translate the structure maps within a DAPM into a presentation matrix of the corresponding persistence module, with work by Skraba and Vejdemo-Johannson [10] regarding the computation of a graded Smith Normal Form, that allows us to derive an interval basis.

Structure of the chapter

The first section reviews some concepts about persistence modules, not necessarily made by the homology of some space, and introduce our notion of interval basis in connection with the Structure Theorem. The following section introduces the main algorithm (alg. 4.3) to compute an interval basis of a general persistence module. Next, we provide a specialized algorithm to deal with real coefficients, where characteristic is zero and we can leverage scalar products. The treatment then moves to an improved version of the general algorithm, with a reduced computational complexity. The following section presents the interplay between interval bases and the primary decomposition of a module, where we provide another approach, more algebraic in flavour, to the computation of said bases. Then, we move to a detailed evaluation of the computational complexity of what presented so far. In what follows, we give a constructive method, acting independently over each step (alg. 4.8) and structure map (alg. 4.9), for constructing a k-persistent homology module. Moreover, we see how our constructive method can be adapted to the case of harmonic persistence module where each step is obtained as the kernel of the Hodge Laplacian operator, and each structure map is easily obtained via alg. 4.10.

Preliminaries

Recall that we define a (discrete, algebraic) persistence module $\mathcal{M}$ as pair $\{M_i, \varphi_i\}_{i\in\mathbb{N}}$ consisting of, for any $i \in \mathbb{N}$:

- a finite-dimensional $\mathbb{F}$-vector space M_i called i^{th}*−step*
- a linear map $\varphi_i : M_i \longrightarrow M_{i+1}$ called i^{th}*−structure map*, such that there exists $n \in \mathbb{N}$ for which φ_i is an isomorphism for all $i \geq n$.

We denote by $\varphi_{i,j} : M_i \to M_j$ with $i < j$ the composition $\varphi_{j-1} \circ \ldots \varphi_i$.
Our definition corresponds to what is normally referred to as finite type persistence module [116].

By the fundamental theorem, any persistence module $\mathcal{M}$ can be thought of as a graded module $\bigoplus_i M_i$ over the ring of polynomials $\mathbb{F}[x]$, or a graded $\mathbb{F}[x]$-module, for short. Indeed, the homogeneous part of degree i is the step M_i and the action over M_i under the indeterminate x is given by applying the structure map φ_i. Being that correspondence an equivalence of categories, we can transpose to persistence modules some notions which apply to graded modules, such as isomorphisms, homogeneous elements, direct sums, generators, and submodules.

Let $I(v) = \{I_i(v), \psi_i(v)\}_{i\in\mathbb{N}}$ be the submodule of $\mathcal{M}$ generated by $v \in M_b$ (i.e., v is homogeneous in M_b of degree b). Observe that we have

$$I_i(v) = \begin{cases} \langle \varphi_{b,i}(v) \rangle & \text{if } i \geq b, \\ 0 & \text{otherwise,} \end{cases} \qquad \psi_i(v) = \begin{cases} \varphi_i|_{\langle \varphi_{b,i}(v) \rangle} & \text{if } i \geq b, \\ 0 & \text{otherwise,} \end{cases}$$

where the brackets $\langle \cdot \rangle$ denotes the $\mathbb{F}$-linear space spanned by their argument.

Now, define an *(integer) interval* $[b,d]$ with $b \leq d$ to be the limited set $\{b, b+1, \dots, d\}$ and $[b,\infty]$ to be the unlimited set $\{b, b+1, \dots\}$. The *interval module* $I_{[b,d]}$ relative to the interval $[b,d]$, possibly with $d = \infty$, is the persistence module $\{I_i, \psi_i\}_{i\in\mathbb{N}}$ such that

$$I_i = \begin{cases} \mathbb{F} & \text{if } b \leq i \leq d, \\ 0 & \text{otherwise,} \end{cases} \qquad \psi_i = \begin{cases} \mathrm{id}_{\mathbb{F}} & \text{if } b \leq i < d, \\ 0 & \text{otherwise.} \end{cases}$$

Remark 2. Fix a degree b. For each $v \in M_b$, there exists $d \in \mathbb{N}$ or $d = \infty$ such that

$$I(v) \cong I_{[b,d]}.$$

Indeed, each step in $I(v)$ is either isomorphic to the vector space $\mathbb{F}$ or to 0 and the structure maps are either isomorphisms or the null map. If there exists an integer r, such that $I_r(v) = 0$, we take d to be the minimum of such r's. Otherwise, we set $d = \infty$.

Definition 62. Given a persistence module $\mathcal{M} = \{M_i, \varphi_i\}_{i\in\mathbb{N}}$, a finite family $\{v_1, \dots, v_N\} \subseteq \bigoplus_i M_i$ of homogeneous elements is an *interval basis* for $\mathcal{M}$ if and only if

$$\bigoplus_{m=1}^{N} I(v_m) = \mathcal{M}$$

.

Proposition 3. *Every persistence module $\mathcal{M} = \{M_i, \varphi_i\}_{i\in\mathbb{N}}$ admits an interval basis $\{v_1, \dots, v_N\} \subseteq \bigoplus_i M_i$.*

Proof. The existence of an interval basis for each $\mathcal{M}$ follows from the interval decomposition corresponding to the Structure Theorem [109] for finitely generated graded $\mathbb{F}[x]$-modules. Indeed, the interval decomposition implies that $\mathcal{M}$ decomposes into a direct sum of interval modules of the form

$$\mathcal{M} \cong \bigoplus_{m=1}^{N} I_{[b_m,d_m]}, \tag{4.3}$$

where the intervals $[b_m, d_m]$ with $b_m \leq d_m \leq \infty$ are uniquely determined up to re-orderings. Let $\Psi = \{\Psi_i\}_{i\in\mathbb{N}} : \oplus_{m=1}^{N} I_{[b_m,d_m]} \longrightarrow \mathcal{M}$ be the graded isomorphism of the interval decomposition in 4.3. Then, for each summand $I_{[b_m,d_m]}$, the map Ψ_{b_m} detects a vector $v_m \in M_{b_m}$. By Remark 2, we have that $I(v_m) \cong I_{[b_m,r_m]}$ for some $b_m \leq r_m \leq \infty$. Observe now that, for all indexes $i \in \mathbb{N}$, the decomposition isomorphisms satisfies $\varphi_i \circ \Psi_i = \Psi_{i+1} \circ \psi_i$, where ψ_i is the structure map of $I_{[b_m,d_m]}$. This implies that $r_m = d_m$ for all indices $i \in \mathbb{N}$. □

Formula (4.3) defines the *barcode* $B(\mathcal{M})$ as the multiset of pairs $\{(b_m,d_m)\}_{m=1}^{N}$, i.e., the collection of pairs in the formula counted with multiplicity. The pairs of the kind (b,∞) are called *essential pairs*.

An alternative persistence module decomposition is the one that obtains a so-called *coherent basis*.

Definition 63. Given a persistence module $\{M_i,\varphi_i\}_{i=0}^{n}$, a *coherent basis* is given by a set of bases $\left\{\mathcal{B}_i = \{v_1^i,\dots,v_{d_i}^i\}\right\}_{i=0}^{n}$, one for each space M_i, such that for any $i = 0,\dots,n$ it holds,

$$\left(\varphi_i(v_k^i) = v_{i_k}^{i+1} \in \mathcal{B}_{i+1} \vee \varphi_i(v_k^i) = 0\right) \quad \forall v_k^i \in \mathcal{B}_i.$$

In words, a coherent basis is a choice of a basis for each step of the module such that the structure maps are block identity matrices.

Example 3. Consider a persistence module $\{M_i,\varphi_i\}_{i=0}^{n}$ where all φ_i's are injective. Take any basis of M_0 as $\mathcal{B}_0$. Suppose inductively that we have already obtained $\mathcal{B}_i$ For some $i = 0,\dots,n-1$, since φ_i is injective, $\varphi_i(\mathcal{B}_i)$ defines a basis for a subspace B_{i+1} in M_{i+1}. Take any basis completion of B_{i+1} in M_{i+1} as $\mathcal{B}_{i+1}$. Hence, $\{\mathcal{B}_i\}_{i=0}^{n}$ is a coherent basis for $\{M_i,\varphi_i\}_{i=0}^{n}$.

Remark 3. Given an interval basis, we get an induced coherent basis.

Proof. An interval basis generates each step's basis $\mathcal{B}_i$ by simply considering the images of each vector v of the interval basis across the maps φ_{ji} with $j = \deg v \leq i$. □

Decomposing a persistence module via an interval basis consists in retrieving, given a persistence module $\mathcal{M}$, an interval basis $v_1,\dots,v_N$ with N counting the number of interval modules in the interval decomposition of Definition definition 62.

In the following section, we present Algorithm 4.3 for addressing the decomposition as stated above. In order to do so from now on, *we restrict to persistence modules* $\mathcal{M} = \{M_i, \varphi_i\}$ *admitting no essential pairs* in the decomposition. Notice that this is not restrictive since we deal with persistence modules of finite type. This means that there exists an integer n such that φ_m is an isomorphism for all $m \geq n$. By setting $\varphi'_m = 0$ for all $m \geq n+1$, we slightly modify our persistence module to get $\mathcal{M}' = \{M_i, \varphi'_i\}$ in such a way that the interval decomposition admits no essential pairs. We notice that the interval modules $I_{[b,d]}$ with $d \leq n$ are preserved. Whereas, an interval module $I_{[b,\infty]}$ decomposing $\mathcal{M}$ corresponds to an interval module $I_{[b,n+1]}$ decomposing $\mathcal{M}'$. Hence, our working assumption is not restrictive in practice.

Decomposition of persistence modules

Consider a persistence module $\{M_i, \varphi_i\}_{i=1}^n$, as described in the previous section. In particular, we suppose without loss of generality that the module is finite and that the last map $\varphi_n : M_n \to M_{n+1}$ is the null one. Denote with m_i the dimension of the space M_i and with r_i the dimension of Im (φ_{i-1}). For each i there is a flag of vector sub-spaces of M_i given by the kernel of the maps $\varphi_{i,j}$:

$$0 \subseteq \ker(\varphi_{i,i+1}) \subseteq \ker(\varphi_{i,i+2}) \subseteq \cdots \subseteq \ker(\varphi_{i,n+1}) = M_i. \tag{4.4}$$

where the last equality holds by the assumption above.
Denote for simplicity each space $\ker(\varphi_{i,j})$ as V^i_j.
An *adapted basis* for the flag in M_i is given by a set of linearly independent vectors $\mathcal{V}^i = \{v_1, \ldots, v_{m_i}\}$, and an index function $J : \mathcal{V}^i \to \{1, \ldots, n-i+1\}$, such that

$$V^i_{i+s} = \langle \{v \in \mathcal{V}^i \mid J(v) \leq i+s\} \rangle \quad \forall s,\ 1 \leq s \leq n-i+1. \tag{4.5}$$

In words, an adapted basis is an ordered list of vectors in M_i such that for every j, the first $\dim V^i_j$ vectors are a basis of V^i_j (where we assume an empty list is a basis of the trivial space).

Lemma 3. *Notice that, without loss of generality, it is possible to choose an adapted basis of M_i in such a way that it contains as a subset a basis of Im (φ_{i-1}).*

Proof. Let us consider an adapted basis $\mathcal{V}^i = (t_1, \dots, t_{m_i})$ for the flag of kernels in M_i, with the vectors $t_1, \dots, t_{m_i}$ ordered by index function J. We construct the desired basis explicitly: set $\underline{\mathcal{V}}^i = \{t_1\}$. For every $s = 2, \dots, m_i$, if $t_s \notin \langle \underline{\mathcal{V}}^i \rangle + \mathrm{Im}\ (\varphi_{i-1})$ add the vector t_s to $\underline{\mathcal{V}}^i$. Otherwise,it must hold that $t_s = \sum_{a<s} \lambda_a t_a + x$ with $x \in \mathrm{Im}\ (\varphi_{i-1})$. Then we add to $\underline{\mathcal{V}}^i$ the vector $x = t_s - \sum_{a<s} \lambda_a t_a$. In this way $\underline{\mathcal{V}}^i$ is another adapted basis, and the elements added by the second route form a basis of $\mathrm{Im}\ (\varphi_{i-1})$. $\square$

From now on, therefore, we shall assume that each basis $\mathcal{V}^i$ is in the form of lemma 3. Let us introduce two subspaces of $\langle \mathcal{V}^i \rangle$: it holds that $\langle \mathcal{V}^i \rangle = \langle \mathcal{V}^i_{\mathrm{Birth}} \rangle \oplus \langle \mathcal{V}^i_{\mathrm{Im}} \rangle$, where $\mathcal{V}^i_{\mathrm{Im}}$ is the subset of $\mathcal{V}^i$ made of a basis of $\mathrm{Im}\ \varphi_{i-1}$, and $\mathcal{V}^i_{\mathrm{Birth}}$ is its complement.

Our objective is to construct a basis of the whole persistence module by leveraging the adapted bases at each step i.

Definition 64. Let us define $\mathscr{V} := \bigcup_i \mathcal{V}^i_{\mathrm{Birth}}$.

So, set $\mathscr{V}$ is made up of the elements of the adapted basis in each degree i that are not elements of $\mathrm{Im}\ (\varphi_{i-1})$. In the following, we prove that $\mathscr{V}$ is in fact an interval basis for $\{M_i, \varphi_i\}_{i=1}^n$.

Lemma 4. *For any $i < j$, the map $\varphi_{i,j}|_T$ restriction of the map $\varphi_{i,j}$ to the subspace $T = \langle \{v \in \mathcal{V}^i \mid J(v) > j\} \rangle$ is an injection.*

Proof. By definition of T, it holds $M_i = \ker(\varphi_{i,j}) \oplus T$. If the restriction of $\varphi_{i,j}$ onto T were not injective, then T and $\ker(\varphi_{i,j})$ would have nontrivial intersection. This is a contradiction. $\square$

Lemma 5. *For any $i < j \in \mathbb{N}$, it holds*

$$\varphi_{i,j}\left(\langle \mathcal{V}^i_{Birth} \rangle\right) \bigcap \varphi_{i,j}\left(\langle \mathcal{V}^i_{Im} \rangle\right) = \{0\}. \tag{4.6}$$

Proof. Suppose that the intersection of the two considered spaces contains a nonzero vector u:

$$0 \neq u = \varphi_{i,j}\left(\sum_{v_k \in \mathcal{V}^i_{\mathrm{Birth}}} \lambda_k v_k\right) = \varphi_{i,j}\left(\sum_{w_l \in \mathcal{V}^i_{\mathrm{Im}}} \mu_l w_l\right).$$

Denote by u_{B} and u_{I} the vectors

$$u_{\mathrm{B}} = \sum_{\substack{J(v_k)>j \\ v_k \in \mathcal{V}^i_{\mathrm{Birth}}}} \lambda_k v_k, \quad u_{\mathrm{I}} = \sum_{\substack{J(w_l)>j \\ w_l \in \mathcal{V}^i_{\mathrm{Im}}}} \mu_l w_l.$$

It holds $u = \varphi_{i,j}(u_{\mathrm{B}}) = \varphi_{i,j}(u_{\mathrm{I}})$, since all the elements v such that $J(v) \leq j$ belong to $\ker(\varphi_{i,j})$. Then, u is the image through $\varphi_{i,j}$ of an element of $T = \langle\{v \in \mathcal{V}^i \mid J(v) > j\}\rangle$. On the other hand also the difference $u_{\mathrm{B}} - u_{\mathrm{I}}$ belongs to the same space and is mapped to zero by $\varphi_{i,j}$. The restriction of $\varphi_{i,j}$ to T is injective because of lemma 4, therefore it must be $u_{\mathrm{B}} - u_{\mathrm{I}} = 0$. Since, $\langle\mathcal{V}^i\rangle = \langle\mathcal{V}^i_{\mathrm{Birth}}\rangle \oplus \langle\mathcal{V}^i_{\mathrm{Im}}\rangle$, it must be $u_{\mathrm{B}} = u_{\mathrm{I}} = 0$, hence $u = 0$.

□

Theorem 7. *The set $\mathscr{V}$ is an interval basis for the persistence module M.*

Proof. Say that $\mathscr{V} = \{v_1, \dots, v_N\}$. Each vector v_j in the set $\mathscr{V}$ induces an interval module $I(v_j)$. We want to show that $M = \bigoplus_{j=1}^N I(v_j)$. To do so, let us see that for each $0 \leq i \leq n$, the space M_i is exactly $\oplus_{j=1}^N I_i(v_j)$. By construction we know that

$$M_i = \langle\{v \in \mathcal{V}^i \mid v \notin \mathrm{Im}\,(\varphi_{i-1})\}\rangle \,\oplus\, \mathrm{Im}\,(\varphi_{i-1}) = \bigoplus_{\substack{v \in \mathscr{V} \\ \deg v = i}} I_i(v) \,\oplus\, \mathrm{Im}\,(\varphi_{i-1}). \quad (4.7)$$

All we have to show is that $\mathrm{Im}\,(\varphi_{i-1})$ can be written as $\bigoplus_{\substack{v \in \mathscr{V} \\ \deg v < i}} I_i(v)$. At first we will see that it holds $\mathrm{Im}\,(\varphi_{i-1}) = +_{\substack{v \in \mathscr{V} \\ \deg v < i}} I_i(v)$. By definition, it holds $+_{\substack{v \in \mathscr{V} \\ \deg v < i}} I_i(v) \subseteq \mathrm{Im}\,(\varphi_{i-1})$. On the other hand, $\mathrm{Im}\,(\varphi_{i-1}) \subseteq +_{\substack{v \in \mathscr{V} \\ \deg v < i}} I_i(v)$. We can see it by induction. Let us consider $M_0 = \langle\mathcal{V}^0\rangle$, where none of the elements of $\mathcal{V}^0$ belongs to $\mathrm{Im}\,(\varphi_{-1}) = \{0\}$. Then, $\mathrm{Im}\,(\varphi_0) \subseteq +_{\substack{v \in \mathscr{V} \\ \deg v = 0}} I_1(v)$. Suppose by induction that for any $k-1$ it holds $\mathrm{Im}\,(\varphi_{k-2}) \subseteq +_{\substack{v \in \mathscr{V} \\ \deg v < k-1}} I_{k-1}(v)$. Then, since $M_{k-1} = \langle\{v \in \mathcal{V}^{k-1} \mid v \notin \mathrm{Im}\,(\varphi_{k-2})\}\rangle \,\oplus\, \mathrm{Im}\,(\varphi_{k-2})$, it holds

$$\mathrm{Im}\,(\varphi_{k-1}) = +_{\substack{v \in \mathcal{V}^{k-1} \\ v \notin \mathrm{Im}\,(\varphi_{k-2})}} I_k(v) + \varphi_{k-1}(\mathrm{Im}\,(\varphi_{k-2})) \quad (4.8)$$

Therefore, by the induction hypothesis

$$\mathrm{Im}\,(\varphi_{k-1}) \subseteq +_{\substack{v \in \mathscr{V} \\ deg v < k}} I_k(v).$$

Now that we have shown that Im $(\varphi_{i-1}) = +_{\substack{v\in\mathscr{V}\\ \deg v<i}} (I(v))_i$, it remains to see that this sum is direct. Suppose to have a non trivial combination $w_1 + \cdots + w_k = 0$, where each w_q belongs to $I_i(v_{t_q})$. Suppose that the $w_1, \dots, w_k$ are ordered according to the degree of the element v_{t_q} that generates the interval module they belong to. Let us say that these elements have a maximum degree $l < i$. Then, it holds

$$w_1 + \cdots + w_k = \varphi_{l,i}(x) + \sum_{v_{t_r}\,|\,\deg v_{t_r}=l} \lambda_r \varphi_{l,i}(v_{t_r}),$$

where $x \in$ Im (φ_{l-1}). Because of lemma 5, it must be

$$\varphi_{l,i}(x) = \sum_{v_{t_r}\,|\,\deg v_{t_r}=l} \lambda_r \varphi_{l,i}(v_{t_r}) = 0.$$

On the other hand we also assumed that the w_q are different from zero, therefore the index $J(v_{t_q})$ of the vectors in the adapted basis has to be greater than $i-l$. Hence, because of lemma 4, it holds that $\sum_{v_{t_r}|\deg v_{t_r}=l} \lambda_r \varphi_{l,i}(v_{t_r}) = 0$ implies $\sum_{v_{t_r}|\deg v_{t_r}=l} \lambda_r v_{t_r} = 0$. Since the $\{v_{t_r} | \deg v_{t_r} = l\}$ are linearly independent it must be $\lambda_r = 0$ for any r, and therefore $w_r = 0$. The same idea can be repeated for all the previous elements $w_1, \dots, w_s$, coming from interval modules generated by vectors with degree less than l. Since there are finitely many vectors this process has an end and it shows that all the vectors $w_1, \dots, w_k$ are 0 and the sum is direct. □

We now provide an explicit construction for set $\mathscr{V}$. To do so, we must first obtain sets $\mathcal{V}^i$.

Remark 4. Notice that the construction of each $\mathcal{V}^i$ is independent from the others. Therefore they can be computed simultaneously.

Construction of $\mathcal{V}^i_{\mathbf{Birth}}$

We first recall that a simple basis extension algorithm is given by the the procedure described in the following alg. 4.1. The set $\mathcal{W}$ is ordered, and its elements are added to $\mathcal{U}$ in their ascending order in $\mathcal{W}$, so that $\mathcal{U}$ is extended to a basis of $\langle\mathcal{U}\rangle + \langle\mathcal{W}\rangle$. In the following, we refer to the extension of basis $\mathcal{U}$ by the vectors in set $\mathcal{W}$ through alg. 4.1 as $\texttt{bca}(\mathcal{U}, \mathcal{W})$.

Algorithm 4.1: Basis completion algorithm

Input: linearly independent vectors $\mathcal{U} = \{u_1, \dots, u_r\}$, linearly independent vectors $\mathcal{W} = \{w_1, \dots, w_n\}$;
Result: minimal set of vectors $w_{i_1}, \dots, w_{i_p} \notin <\mathcal{U}>$ such that $<\mathcal{U} \cup \{w_{i_1}, \dots, w_{i_p}\}> = <\mathcal{U} \cup \mathcal{W}>$

```
R = {};
for i=1,..., n do
    if rank U < rank(U ∪ {w_i}) then
        U = U ∪ {w_i};
        R = R ∪ {w_i};
    end
end
return R
```

We now give a general algorithm to construct the set $\mathcal{V}^i_{\text{Birth}}$ for a given M_i of the persistence module.

To find an adapted basis $\mathcal{V}^i$ we need only to iteratively complete a basis of $\ker(\varphi_{i,j})$ to a basis of $\ker(\varphi_{i,j+1})$, using for example the algorithm described above. In general, the basis obtained through alg. 4.1 will not contain a basis of the space Im (φ_{i-1}), so the procedure does not match the construction in lemma 3. We show below that this is not strictly necessary, so we can save computations without hindering the correctness of the algorithm. In fact, we obtain the set $\mathcal{V}^i_{\text{Birth}}$ applying the reduction algorithm, using as inputs any basis of Im (φ_{i-1}) and the adapted basis $\mathcal{V}^i$.

This is equivalent to using the construction of lemma 3 and then discarding the elements that belong to Im (φ_{i-1}). Consider the ordered set $\mathcal{V}^i = \{v_1, \dots, v_{m_i}\}$ and a set $\mathcal{U} = \{u_1, \dots, u_{r_i}\}$, basis of Im (φ_{i-1}). Vector v_j is modified by the procedure in lemma 3 if it can be written as a linear combination of the vectors $\{u_1, \dots, u_{r_i}, v_1, \dots, v_{i-1}\}$. If that is the case, it is modified so that it belongs to $\mathcal{V}^i_{\text{Im}}$, and then it is not included in $\mathcal{V}^i_{\text{Birth}}$. In the same way, alg. 4.1 discards such vectors without performing further calculations. The vectors belonging to $\mathcal{V}^i_{\text{Birth}}$ are the same in both procedures, hence the result is valid.

The full procedure to construct $\mathcal{V}^i_{\text{Birth}}$ from M_i and the structure maps is described in alg. 4.2.

Algorithm 4.2: Single step decomposition

Input: map $\varphi_{i-1} : M_{i-1} \to M_i$, maps $\{\varphi_j : M_j \to M_{j+1})\}, i \leq j \leq N$;
Result: $\mathcal{V}^i_{\text{Birth}}$ and associated index function J
Take a basis $\mathcal{U} = \{u_1, \dots, u_k\}$ of Im (φ_{i-1});
$k := \dim \text{Im } (\varphi_{i-1})$;
$R := \text{Id} : M_i \to M_i$;
$r := \text{rank}(R)$;
$\mathcal{V}^i_{\text{Birth}} = \{\}$;
for $s = 0, \dots, N-i$ **do**
  $R = \varphi_{i+s} \cdot R$;
  $r' = \text{rank} R$;
  if $r' < r$ **then**
    $B =$ basis of $\ker(R)$;
    $v^s_1, \dots, v^s_{\rho_s} = \text{bca}(\mathcal{U}, B)$;
    $\mathcal{U} = \mathcal{U} \cup \{v^s_1, \dots, v^s_{\rho_s}\}$;
    $\mathcal{V}^i_{\text{Birth}} = \mathcal{V}^i_{\text{Birth}} \cup \{v^s_1, \dots, v^s_{\rho_s}\}$, $J(v^s_1) = \cdots = J(v^s_{\rho_s}) = s+1$;
    $r = r'$;
  end
  if $r = 0$ ***or*** $|\mathcal{V}^i_{Birth}| + k = \dim M_i$ **then**
    break ;
  end
end
return $\mathcal{V}^i_{Birth}$, $J : \mathcal{V}^i_{Birth} \to \{1, \dots, N-i+1\}$.

In the following, we refer to the construction of $\mathcal{V}^i_{\text{Birth}}$ through alg. 4.2 as $\texttt{ssd}(M_i)$. Once the decomposition of each space is performed, it is immediate to assemble the interval basis $\mathscr{V}$. Further, we can read the persistence diagram of module $\{M_i, \varphi_i\}_i$ off of the interval basis by storing the indices of appearance and death of its elements, without increasing the computational cost. This is the content of alg. 4.3, which summarizes the procedures introduced so far into a single routine that takes a persistence module and returns its interval basis and persistence diagram.

Algorithm 4.3: Persistence module decomposition

Input: persistence module $\{M_i, \varphi_i\}_{i=1}^n$;
Result: interval basis $\{v_i^s\}$ and persistence diagram
$\varphi_0 :=$ empty matrix with $\dim M_0$ rows and 0 columns;
$\varphi_{n+1} :=$ empty matrix with 0 rows and $\dim M_n$ columns;
$PB = \{\}$;
$PD = \{\}$;
for $b = 1, \dots, n+1$ **do**
    $\mathcal{V}^i_{\text{Birth}}, J = \text{ssd}(\varphi_{b-1}, \{\varphi_j\}_{j \geq b})$;
    if $\mathcal{V}^i_{Birth}$ *is not empty* **then**
        Add to the interval basis PB the vectors in $\mathcal{V}^i_{\text{Birth}}$;
        Update the persistence diagram with the points $(b, J(v) + b)$ for $v \in \mathcal{V}^i_{\text{Birth}}$;
    end
end
return *Interval basis, persistence diagram*

We refer to the decomposition of alg. 4.3 as $\texttt{pmd}(M)$.

Example 4. Consider the following $\mathbb{R}$-persistence module

$$0 \xrightarrow[\begin{pmatrix}0\end{pmatrix}]{\varphi_0} \mathbb{R} \xrightarrow[\begin{pmatrix}1\\0\end{pmatrix}]{\varphi_1} \mathbb{R}^2 \xrightarrow[\begin{pmatrix}1 & 1\end{pmatrix}]{\varphi_2} \mathbb{R} \xrightarrow[\begin{pmatrix}0\end{pmatrix}]{\varphi_3} 0$$

Where the matrices below each arrow represent the map above it in the canonical bases. We showcase the procedure of alg. 4.3 and compute its interval basis. Notice that this example matches the persistence module generated by persistent homology of the filtration in fig. 4.1, whose interval basis is depicted in fig. 4.3.

For $i = 0, 1, 2, 3$ we need to compute $\mathcal{V}^i_{\text{Birth}}$.
Clearly φ_0 is the null map, so the flag for the first step is trivial and $\mathcal{V}^0_{\text{Birth}}$ is empty.

For $i = 1$ we have Im $(\varphi_{i-1}) = 0$ and $\ker(\varphi_{1,2}) = \ker(\varphi_{1,3}) = 0$, so $\mathbb{R} = \ker(\varphi_{1,4})$. By `ssd` we extend a basis of Im (φ_0) (which is empty) to a basis of $\mathbb{R}$, which yields vector 1. Then $\mathcal{V}^1_{\text{Birth}} = \{1\}$ with persistence pair $(1, 4)$.

For $i = 2$ we have Im $(\varphi_{i-1}) = \langle \begin{pmatrix} 1 \\ 0 \end{pmatrix} \rangle$. Furthermore $\ker(\varphi_{2,3}) = \langle \begin{pmatrix} 1 \\ -1 \end{pmatrix} \rangle$, so we extend the basis of Im (φ_1) against the basis of $\ker(\varphi_{2,3})$ obtaining set $\{ \begin{pmatrix} 1 \\ -1 \end{pmatrix}, \begin{pmatrix} 1 \\ 0 \end{pmatrix} \}$, which spans $\mathbb{R}^2$, so `ssd` terminates setting $\mathcal{V}^2_{\text{Birth}} = \{ \begin{pmatrix} 1 \\ -1 \end{pmatrix} \}$ with persistence pair $(2, 3)$.

For $i = 3$ we have Im $(\varphi_{i-1}) = \mathbb{R}$, so $\mathcal{V}^3_{\text{Birth}}$ is empty.

Finally, the interval basis is $\mathscr{V} = \{1, \begin{pmatrix} 1 \\ -1 \end{pmatrix} \}$, with persistence diagram $PD = \{(1, 4), (2, 3)\}$.

The case of real coefficients

In case we use the field $\mathbb{R}$ in the persistence module, we can specialise the decomposition of the space described in the previous paragraph. We will use the following notation: given a matrix A with m rows and n columns, $A[:, i]$ denotes the i^{th} column of the matrix , whereas $A[:, : i]$ denotes the submatrix given by the first i columns of A. The same notation is used on the first arguments in the parenthesis to denote operations on rows. We will make use of this simple result in linear algebra.

Lemma 6. *Given three vector spaces V_1, V_2, and V_3 over $\mathbb{R}$ and two linear maps $\psi_1 : V_1 \to V_2$ and $\psi_2 : V_2 \to V_3$ it holds*

$$\ker(\psi_2 \circ \psi_1) = \ker(\psi_1) \oplus \ker\left(\psi_2 \circ \psi_1|_{(\ker(\psi_1))^\perp}\right).$$

Proof. Let x be an element of $\ker(\psi_2 \circ \psi_1)$. It can be written uniquely as $x = v + w$, with $v \in \ker(\psi_1)$ and $w \in (\ker(\psi_1))^\perp$. Since $(\psi_2 \circ \psi_1)(v + w) = 0$ and $v \in \ker(\psi_1)$, it must be $\psi_2(\psi_1(w)) = 0$, therefore $w \in \ker(\psi_2 \circ \psi_1)$. Then, w belongs to $\ker\left(\psi_2 \circ \psi_1|_{(\ker \psi_1)^\perp}\right)$ and the statement follows. $\square$

Fix M_0, and suppose that $\varphi_n = 0$.

For each M_i, denote with d_i the number $\dim M_i$. Consider φ_0 and decompose it via the SVD decomposition in $\varphi_0 = U_0 S_0 V_0^T$. If $r_0 = \operatorname{rank}\varphi_0$, then $k_0 = d_0 - r_0$ is the dimension of $\ker\varphi_0$. Notice that S_0 is a matrix $d_1 \times d_0$ with non-zero elements only on the first r_0 positions on the main diagonal. Therefore, if e_i is the i^{th} element of the canonical basis of $\mathbb{R}^{d_0}$, with $r_0 < i \leq d_0$, then $\varphi_0 V_0 e_i = U_0 S_0 e_i = 0$. Then, a basis of $\ker\varphi_0$ is given by the vectors $\{V_0 e_{r_0+1}, \dots, V_0 e_{d_0}\}$. The index function J attains the value 1 on all of them. All such vectors will be also in the kernel of the maps $\varphi_{0,j}$ for all $j > 0$. In order to avoid repetitions, it will be considered only the restriction of each $\varphi_{0,j}$ on the orthogonal complement of $\ker\varphi_0$. This operation will not change the result because of lemma 6. To do so, consider the map $\tilde{\varphi}_0 = U_0 \tilde{S}_0$, where $\tilde{S}_0 = S_0[:,: r_0]$, given by the first r_0 columns of S_0. Repeating the same process, it will be considered $m_1 = \varphi_1 \tilde{\varphi}_0$ instead of $\varphi_{0,2}$. Call $d_1 = d_0 - k_0$. Decompose again $m_1 = U_1 S_1 V_1^T$ and call $r_1 = \operatorname{rank} m_1$ and $k_1 = d_1 - r_1 = \dim\ker m_1$. Again, a basis of $\ker m_1$ is given by the vectors $V_1 e_{r_1+1}, \dots, V_1 e_{d_1}$. Recall that this vectors are expressed in the basis $\{V_0[:,1], \dots, V_0[:,r_0]\}$ of $\ker\varphi_0^{\perp}$. To return them in the canonical basis of M_0 it is sufficient to consider the matrix η_0 with d_0 rows and r_0 columns such that $\eta_0[i,j]$ is equal to 1 if $1 \leq i = j \leq r_0$ and 0 otherwise. Then, the vectors in the canonical basis of M_0 are $\{V_0 \eta_0 V_1 e_{r_1+1}, \dots, V_0 \eta_0 V_1 e_{d_1}\}$. In this case the value of the index function for these vectors will be 2. For the general step j, consider $m_j = \varphi_j \tilde{m}_{j-1} = U_j S_j V_j^T$. The adapted basis of M_0 will be updated with the vectors

$$V_0 \eta_0 \dots V_{j-1} \eta_{j-1} V_j e_x, \quad r_j + 1 \leq x \leq d_j, \tag{4.9}$$

and it will be $J(V_0 \eta_0 \dots V_{j-1} \eta_{j-1} V_j e_x) = j + 1$ for every $r_j + 1 \leq x \leq d_j$ Once all the vectors are obtained, as in the general case, it is necessary to complete a basis of $\operatorname{Im}(\varphi_{i-1})$ to a basis of M_0, introducing the vectors in $\mathcal{V}$ in ascending order given by the function J. The resulting vectors will be part of the interval basis.
The procedure is encoded in alg. 4.5, which makes use of the matrix decomposition routine alg. 4.4 and specializes alg. 4.2 to the case of real coefficients. We denote it by $\texttt{ssdR}(M_i)$. Then, the full decomposition of alg. 4.3 can be specialized to the reals by replacing $\texttt{ssd}(M_i)$ with $\texttt{ssdR}(M_i)$.

Algorithm 4.4: Matrix decomposition

Input: matrix A;

Result: Restriction of A on the space orthogonal to its kernel with respect to a basis V of the domain, V matrix whose columns are a basis of the domain of A, $\dim(\ker A)^{\perp}$,$\dim \ker A$

$U,S,V = \text{SVD}(A)$;

$nz = \text{rank}\, S$, $d =$ number of columns of A, $dk = d - nk$;

$R = US[:,:nz]$;

return *R, V, nz, dk*

Algorithm 4.5: single step decomposition on $\mathbb{R}$

Input: map $\varphi_{i-1} : M_{i-1} \to M_i$, maps $\{\phi_j : M_j \to M_{j+1})\}, i \le j \le N$;
Result: Vectors $\mathcal{V}^i_{\text{Birth}}$
$U, S, V = \text{SVD}(\varphi_{i-1})$;
$r := \text{rank}(\varphi_{i-1})$;
$\mathcal{U} = U[:, : r]$ basis of the image of φ_{i-1};
$\mathcal{V}^i_{\text{Birth}} = \{\}$; $lk = 0$;
$R = \text{Id} : M_i \to M_i$;
$d := \dim M_i$;
$V_{tot} = I_d$;
for $s = 0, \dots, N - i$ **do**
    $R = \varphi_{s+1} \cdot R$;
    if *number of rows of* $R = 0$ **then**
        $k :=$ number of columns of R;
        $V = I_k$;
        $nz = 0$;
        $dk = k$;
    else
        $R, V, nz, dk = \text{dec}(R)$;
    end
    $V_{temp} = I_d$, $l = \text{ord}(V)$, $V_{temp}[: l, : l] = V^t$;
    $V_{tot} = V_{tot} \cdot V_{temp}$;
    if $dk > 0$ **then**
        $\mathcal{T} = \text{bca}(\mathcal{U}, V_{tot}[:, d - lk - dk : \ d - lk])$;
        $\mathcal{U} = \mathcal{U} \cup V_{tot}[:, d - lk - dk : \ d - lk]$;
        $\mathcal{V}^i_{\text{Birth}} = \mathcal{V}^i_{\text{Birth}} \cup \mathcal{T}$;
        $J(t) = s + 1$ for all $t \in \mathcal{T}$;
        $lk = lk + dk$;
    end
    if $nz = 0$ ***or*** $|\mathcal{V}| + r = d$ **then**
        break;
    end
end
return $\mathcal{V}^i_{Birth}$, J

Improved decomposition algorithm

Algorithm 4.2 can be modified to obtain a computationally improved version.
The modification is based on the observation that the reduction of matrix R can be used to update R for the following step, by restricting it to the non-null columns. This decreases the complexity of all matrix multiplications and reductions involved in the for-cycle, i.e. throughout the subsequent steps of the module.
Moreover, the basis completion procedure can take advantage of this collapse of matrix R, in terms of how the linear independence among kernel generators is checked. Indeed, with the collapsed R it is no longer needed to check new-found vectors against the already stored vectors with lower persistence. The new-found vectors need simply to be checked against the basis of the image of φ_{i-1}, since the independence from the other vectors from the kernel is ensured by the deletion of the zero columns from the collapsed matrix R of the previuos step.
This change alters significantly the complexity of the single step decomposition. We sketch give the modified algorithm below. The implicit procedure *ColumnReduction* simply reduces matrix R, storing the change of basis matrix C, ignoring columns that had become zero and highlighting the indices of the new columns that become zero.

Algorithm 4.6: Improved Single step decomposition

Input: map $\varphi_{i-1} : M_{i-1} \to M_i$, maps $\{\varphi_j : M_j \to M_{j+1})\}, i \leq j \leq N$;
Result: $\mathcal{V}^i_{\text{Birth}}$ and its index function J

```
Reduce φ_{i-1} and find a basis 𝒰 = {u_1,...,u_k} of Im (φ_{i-1});
k := dim Im (φ_{i-1});
R ← Id : M_i → M_i;
C ← Id : M_i → M_i;
r ← dim(M_i);
𝒱^i_Birth ← {};
inds, newInds ← {};
for s = 0,...,N − i do
    R ← φ_{i+s} · R ;
    R, C, newInds ← ColumnReduction(R, C, inds);
    r' ← rank R = r − |newInds|;
    if r' < r then
        B ← basis of ker(R) = {Ce_i, i ∈ newInds};
        B_new ⊆ B ← bca(𝒰, B);
        𝒰 ← 𝒰 ∪ B_new;
        𝒱^i_Birth ← 𝒱^i_Birth ∪ B_new;
        for v ∈ B_new do
            J(v) ← s + 1;
        end
        r ← r';
        inds ← inds ∪ newInds;
    end
    if r = 0 or |𝒱^i_Birth| + k = dim M_i then
        break ;
    end
end
return 𝒱^i_Birth, J
```

Interval bases and the primary decomposition

In this section we propose an alternative method to compute an interval basis, based on a suitable reduction of a presentation matrix. It is based on a combination of two technical ingredients: first, the construction of "any" presentation matrix for our persistence module. This is done in a way that, to our knowledge, was first explicitly envisaged in a technical passage of [12]. Next, by the reduction of this presentation matrix into a nearly-diagonal form (i.e. diagonal up to a permutation of rows), so that each syzygy in the module is *pure*, i.e. concerns exactly one generator. This is done by leveraging a graded Smith Normal Form reduction that was, again to the best of our knowledge, first presented in [10].
In the process of doing so, we also propose a modification of this algorithm, based on the following observation: while theoretically the reduction of an $\mathbb{F}[x]$-module is performed by operating on a matrix with coeffiecients in the PID $\mathbb{F}[x]$, in practice we can leverage the grading structure and the explicit knowledge of the degree of each generator and relation, plus the fact that each operation is homogeneous, to only consider operations over $\mathbb{F}$, much in the same way as for the classical algorithm for persistent homology.

In the following, let us frame the problem from the point of view of module presentations. We have a family of maps of $\mathbb{F}$-vector spaces $\mathcal{M} = (M_i, \varphi_i)_i$.
Fix the standard grading on $\mathbb{F}[x]$. Build a graded $\mathbb{F}[x]$-module $\alpha(\mathcal{M}) := \bigoplus_i M_i$, with this given grading, i.e. such that $x^p M_i \subseteq M_{i+p}$. In particular, we define that x acts on a homogenous element m_i by $\varphi_i(m_i)$.

In other words, if we fix the obvious basis for $\bigoplus_i M_i$, we can write the φ_i's together as a map

$$\Phi : \bigoplus_i M_i \to \bigoplus_i M_i$$

So, on any element $m \in \alpha(\mathcal{M})$, multiplication by polynomial $p(x) \in \mathbb{F}(x)$ acts as

$$p(x) \cdot m := p(\Phi)(m)$$

$\alpha(\mathcal{M})$ must fit in the following exact sequence

$$0 \to \ker \mu \overset{i}{\hookrightarrow} \mathbb{F}[x]^g \overset{\mu}{\twoheadrightarrow} \alpha(\mathcal{M}) \to 0$$

which amounts to giving a *presentation* of $\alpha(\mathcal{M})$.
Fixing g generators of the free module, saying the module is *given* amounts to giving homogenous generators of the syzygy module, i.e. a set of homogenous elements of the free module that span Im i. Then

$$\alpha(\mathcal{M}) \cong \mathbb{F}[x]^g \ / \ \mathrm{Im}\, i \cong \mathbb{F}[x]^g \ / \ \ker \mu$$

As per [12], from the φ_i's in the DAPM one can build these *homogeneous* generators of Im i, $\langle r_1, \dots, r_s \rangle$. They can be stored as a $g \times s$ matrix S representing map i.
It then holds

$$\alpha(\mathcal{M}) \cong \mathbb{F}[x]^g \ / \ \mathrm{Im}\, S \cong \mathbb{F}[x]^g \ / \ \langle r_1, \dots r_s \rangle$$

In particular, by the procedure we propose one obtains a matrix S with the same size as Φ, i.e. where $g = s = \dim \bigoplus_i M_i$.
The construction is as follows: assume the module has d steps, each of dimension m_i. Matrix S is nearly a block matrix, although not quite. In particular, call d_i the first index of a generator of the i^{th} step, i.e. $d_i = \sum_{j<i} m_j + 1$. Then for each $i = 1, \dots d-1$, matrix S contains a block on columns from d_i to $d_{i+1} - 1$ and on rows from d_i to $d_{i+2} - 1$. The sub-block of rows from d_i to $d_{i+1} - 1$ (whose diagonal is the main diagonal of S) is a diagonal matrix with all entries equal to $-x$. The off-diagonal sub-block of rows going from d_{i+1} to $d_{i+2} - 1$, which is a $m_{i+1} \times m_i$ matrix, contains the matrix representing φ_i. Outside of these blocks, S is zero. The last block is diagonal and contains no second sub-block, as everything past that point is mapped to zero.
Notice two consecutive blocks overlap between the second sub-block of the first and the first sub-block of the second. This will impact the computational complexity of the reduction procedure. Further details can be found in the proof of lemma 6 of [12].

Definition 65. Given a DAPM, we refer to the construction above as its presentation matrix S.

Notice that the structure of the (sub-) blocks in S determines the degree of each element. In particular, each row can be assigned the degree of the corresponding generator, so that each row between d_i and $d_{i+1} - 1$ (included) has degree i. The same goes for columns: each column represents the action on one of the generators of some map φ_i, which is to send this generator up one degree. Indeed, each column between d_i and $d_{i+1}-1$ is supported on the rows between d_i to $d_{i+2}-1$, and contains $-x$ (an element of degree 1) on a row of degree i and some scalars (elements of degree 0) on rows of degree $i+1$. Hence each column between d_i and $d_{i+1}-1$ has degree $i+1$.

This entails that each element in matrix S is either 0, or has a degree that is equal to the difference between the degrees of its column and the degree of its row.

This is what will allow us to run the whole procedure without symbolic computations. The only genuinely relevant information in the matrix is whether an element is zero or not, because other than that its degree is determined by its position.

Example 5. Consider the same persistence module as in the example 4.

$$0 \underset{\begin{pmatrix}0\end{pmatrix}}{\xrightarrow{\varphi_0}} \mathbb{R} \underset{\begin{pmatrix}1\\0\end{pmatrix}}{\xrightarrow{\varphi_1}} \mathbb{R}^2 \underset{\begin{pmatrix}1 & 1\end{pmatrix}}{\xrightarrow{\varphi_2}} \mathbb{R} \underset{\begin{pmatrix}0\end{pmatrix}}{\xrightarrow{\varphi_3}} 0$$

We ignore the zero maps as they are immaterial to the matrix construction. We say the module has three steps $M_1 = \mathbb{R}$ of degree 1, $M_2 = \mathbb{R}^2$ of degree 2 and $M_3 = \mathbb{R}$ of degree 3. Matrix S is 4×4, and it holds $d_1 = 1,\ d_2 = 2,\ d_3 = 4$. Then matrix S is

$$S = \begin{pmatrix} -x & 0 & 0 & 0 \\ 1 & -x & 0 & 0 \\ 0 & 0 & -x & 0 \\ 0 & 1 & 1 & -x \end{pmatrix}$$

When implemented in practice, the terms $-x$ are substituted by -1, as their degree is implicit by their position.

In general, the presentation obtained via definition 65 is far from minimal, in the sense that several pairs of generator-relation are in excess and can be discarded while maintaining the module fixed.

The graded Smith Normal Form in [10] computes a graded primary decomposition over $\mathbb{F}[x]$ of this module, changing basis for generators and syzygies to obtain new (homogenous) relations $r'_1, \dots, r'_g$ that are, up to reordering, diagonal.

Of these, some are zero, corresponding to free generators of the module. Others are units (invertible in $\mathbb{F}[x]$, i.e. non-zero scalars), which correspond to surplus generators which should be discarded. Finally others are elements of positive degree, corresponding to torsion elements. Then

$$\alpha(\mathcal{M}) \cong \mathbb{F}[x]^g \,/\, \left(\langle r'_1\rangle \oplus \cdots \oplus \langle r'_g\rangle\right) \;\cong\; \mathbb{F}[x] \,/\, \langle r'_1\rangle \oplus \cdots \oplus \mathbb{F}[x] \,/\, \langle r'_g\rangle \;\cong$$

$$\cong \bigoplus_{r'=0} \mathbb{F}[x] \oplus \bigoplus_{\deg r'_i > 0} \mathbb{F}[x] \,/\, \langle r'_i\rangle \,\oplus\, \bigoplus_{r' \text{ unit}} 0$$

The algorithm is as follows:
By *low* of a column we refer to the index of its last (downward) non-zero entry. Notice no column of S is zero in the beginning.

Algorithm 4.7: Graded Smith Normal Form

Input: Matrix S as per definition 65 ;
Result: Matrices $SNF(S)$ and the change of basis matrix R
$R := \mathrm{Id}_{g\times g}$;
for *each column c of S from left to right* **do**
  $l :=$ low of c ;
  For all rows from $l-1$ upward, get 0 in column c by summing row l ;
  Store the row operations in R ;
  For all columns from $c+1$ rightward, get 0 in row l by summing column c ;
end
return $SNF(S), R$

The output of this procedure is the graded Smith Normal Form $SNF(S)$ (which is diagonal up to reordering), alongside the change of basis matrix R, which is a $g \times g$ matrix that refers to the change of basis of the *row space*, i.e. of the generators of the free module.
Knowing the degree structure of $SNF(S)$, which is the same as that of S, we can read the barcode off of it: each row i has a degree $\deg i$. It either participates in a single relation (column) j, of degree $\deg j$, or it is a zero row. In the first case, it generates a submodule of length $\deg j - \deg i$. If this difference is zero, the generator can equivalently be discarded, as it corresponds to a virtual, zero-persistence, pair. In the case of a zero row, it is a free generator.
Matrix R is a row basis change matrix for map $i\ :\ \ker\mu \hookrightarrow \mathbb{F}[x]^g$, i.e. a change of generators in the free module. Each column c in R represents the *old* generator g_c in terms of the *new* generators $g'_1, \ldots, g'_g$. Notice it is invertible.
In its inverse, R^{-1}, it holds that each column c represents the *new* generator g'_c in

terms of the *old* generators $g_1, \dots, g_g$. Those corresponding to a zero-length pair in SNF (an invertible element) can be discarded. The others form an interval basis.

Theorem 8. *The columns of R^{-1} corresponding to non-zero length pairs in SNF form an interval basis for $\mathcal{M}$.*

Proof. That they span $\mathcal{M}$ descends from the Structure theorem. That they do so as a direct sum descends by the diagonal structure of the relations with respect to that basis, which implies that the only linear relations among those generators are of the form $x^p g = 0$. □

Example 6. (continued) From matrix S obtained in example 5, let us compute an interval basis. Algorithm 4.7 applied to S yields

$$SNF(S) = \begin{pmatrix} 0 & 0 & 0 & x^3 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & -x & 0 \\ 0 & 1 & 0 & 0 \end{pmatrix}$$

We see that rows 2 and 4 correspond to surplus generators, as they contain a unit in $SNF(S)$. Row 1 corresponds to a bar born at degree 1, and killed by a relation (column) of degree 4, hence yielding a pair $(1,4)$. Row 3 corresponds to a bar born at degree 2, and killed by a relation of degree 3, hence yielding a pair $(2,3)$.
The change of basis matrix R is

$$R = \begin{pmatrix} -1 & -x & -x & -x^2 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

whose inverse equals itself

$$R^{-1} = \begin{pmatrix} -1 & -x & -x & -x^2 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

Then, column 1 and 3 in this matrix, corresponding to non-zero persistence generators, form an interval basis. They are $-g_1$ and $-xg_1 + g_3$. They are indeed the first cycle to be born (with a minus sign, which is irrelevant), and the difference between the first cycle mapped at the second step and the second cycle. We remark that $xg_1 = g_2$ Notice everything is exactly as in the example 4.

We have implemented this procedure as Python code, as a purely numerical matrix construction and reduction scheme, and plan to render it publicly available soon.

Computational complexity estimates

Here, we give an estimate of the computational complexity of the algorithms presented above. The conclusion that we draw is that the SNF procedure has a complexity that is essentially only dependent on the size of the barcode throughout the steps. The *naive* version of the parallel algorithm has the same order of complexity, even with a worse factor. The *improved* version, instead, has a complexity estimate which does not only depend on the size of the barcode, but also on how the bars are connected. Its complexity is, roughly speaking, output dependent. When exploiting a parallel implementation, the improved algorithm has a complexity that in the worst case equals that of the SNF, and on average outperforms it.

Let us assume that our module has N steps, each called M_i, each having dimension m_i. Assume that, *on average*, each M_i has dimension m. Then, a reasonable estimate of the complexity of the graded Smith normal form is of $O(N\,m^3 + \text{lower order terms})$ [10]. Notice that matrix R is always triangular, as we only combine a row to the rows above it, therefore its inverse is inexpensive to compute and does not add a significant term. Plus, forming matrix S is essentially book-keeping, depending on the implementation virtually cost-less.

Algorithm 4.2, in its basic version, for the generic step M_i has a complexity estimate of $O(m_i^3 + (N-i)(3m_i)^3)$. Therefore, even when computation is fully distributed and each step is analyzed at the same time, the worst-case steps still have a complexity that is of the same order of the full SNF procedure.

The modified version of the single step decomposition, given in alg. 4.6, has an estimate that is output dependent.
First, a column reduction is called once before the for cycle to extract the image of φ_{i-1}, reducing a matrix of size $m_i \times m_{i-1}$.
We observe that inside the for cycle the number of operations depends on the parameter $k_i = m_i - r_i$, where $r_i = \operatorname{rank} \varphi_{i-1}$, and on the variable parameter r_s that counts the number of columns that have not collapsed.
Indeed within the for cycle, for each s:

- A matrix multiplication is called for matrices of sizes $m_{i+1+s} \times m_{i+s}$ and $m_{i+s} \times r_s$, where r_s is the number of columns in R that have not collapsed.
- A column reduction is called for a matrix of size $m_{i+1+s} \times r_s$.
- A basis completion alg. 4.1 (`bca`) is called between a list of r_i vectors and a list of $|\text{newInds}|$ vectors.

Now, the `bca` operation is performed in chunks as s increases, but the sum of the $|\text{newInds}|$ eventually amounts to k_i. By the property of interval generators of remaining linearly independent, the total cost of `bca` for the whole for cycle amounts to that of reducing a list of k_i vectors against a list of r_i vectors, each vector being m_i entries long. The total cost is therefore $O(m_i\ r_i\ k_i) < O(m_i^3)$.

The cost of matrix multiplication between a $m_{i+1+s} \times m_{i+s}$ and a $m_{i+s} \times r_s$ matrix is $O(m_{i+1+s}\ m_{i+s}\ r_s)$. Let us again leverage the assumption that on average all steps have dimension m, and we get $O(m^2 r_s)$. Now, the sum $\sum_s r_s$ is, roughly speaking, the "volume" of all bars born at step M_i, until their death. Denote it by $V(M_i)$. Hence, the total cost of matrix multiplication throughout the for cycle is $O(m^2 V(M_i))$. We can bound this quantity by the case of the rectangular barcode, which is the worst case and where all bars live from the beginning to the end, as $V(M_i) \leq m(N-i)$, obtaining a total cost of $O(m^3(N-i))$.

As for the column reduction, we can again assume an average dimension of m and bound each r_s by m itself, as in the worst case for the fully rectangular barcode, obtaining another estimate of $O(m^3(N-i))$.

Putting all three together along with the column reduction before the for cycle yields an estimate of $O(m^3(2(N-i)+1))$ as the cost of decomposing one step via alg. 4.6 in the worst case. Assuming a parallel implementation, this amounts to the cost of the total procedure alg. 4.3. We see therefore that in the worst case the parallel approach has the same order of complexity as the SNF approach.
Notice that computing the decomposition of an H_0 persistence module, where all generators are born at step zero, could attain a complexity similar to the theretical worst case, when in the parallel pipeline the first job handles the full decomposition, while all the others terminate at once.

However, in the average case, the classes that are spawned at M_i do not fill the whole persistence module, i.e. usually a barcode is relatively sparse and $V(M_i)$ is significantly smaller than $m(N-i)$. As such, in the average case the cost of the parallel approach is closer to $O(m^2\eta + m\eta^2)$ with η an area parameter that we could reasonably estimate as $\eta \sim m$, therefore attaining an advantage with respect to the SNF approach.

In the following section, we shall employ the concepts introduced so far to address the specific case of persistence modules arising from persistent homology. To this aim, we recall a few basic notions about persistent homology.

The k-persistent homology module

A *chain complex* with coefficients in $\mathbb{F}$ is a sequence $C = (C_\bullet, \partial_\bullet)$ of $\mathbb{F}$-vector spaces connected by linear maps with $k \in \mathbb{N}$

$$\dots \xrightarrow{\partial_{k+2}} C_{k+1} \xrightarrow{\partial_{k+1}} C_k \xrightarrow{\partial_k} C_{k-1} \xrightarrow{\partial_{k-1}} \dots \xrightarrow{\partial_2} C_1 \xrightarrow{\partial_1} C_0 \xrightarrow{\partial_0} 0,$$

such that $\partial_{k+1}\partial_k = 0$ for all $k \in \mathbb{N}$. Each $\mathbb{F}$-vector space C_k is called the space of *k-chains*.

A chain complex can be constructed from a *simplicial complex*.

The subspace $Z_k = \ker(\partial_k)$ is called the space of *k-cycles*. The subspace $B_k = \mathrm{Im}\ (\partial_{k+1})$ is called the space of *k-boundaries*. The condition $\partial_{k+1}\partial_k = 0$ ensures that

$$B_k \subseteq Z_k, \qquad \text{for all } k \in \mathbb{N}.$$

The quotient space $H_k = Z_k/B_k$ is called the *k-homology space*.

A *chain map* $f : (C_\bullet, \partial_\bullet^C) \longrightarrow (D_\bullet, \partial_\bullet^D)$ is a collection of linear maps $f_k : C_k \longrightarrow D_k$ such that

$$f_k \partial_{k+1}^C = \partial_{k+1}^D f_{k+1}, \text{ for all } k \in \mathbb{N}. \tag{4.10}$$

A chain map induces a linear map of homology spaces. Indeed, for each k-cycle z in C_k or D_k, we write $[z]_C$ or $[z]_D$ for its projection onto the homology space H_k^C or H_k^D, respectively. By Equation (4.10), we get that $f_k(B_k^C) \subseteq B_k^D$. Hence, we get induced a map $\tilde{f}_k : H_k^C \longrightarrow H_k^D$ defined by

$$\tilde{f}_k([z]_C) = [f_k(z)]_D. \tag{4.11}$$

This property is called *functoriality* of the homology construction.

The *k-persistent homology module* with coeffients in $\mathbb{F}$ is the persistence module obtained from a sequence of chain maps

$$0 \xrightarrow{f_{(0)}} C_{(1)} \xrightarrow{f_{(1)}} \dots \longrightarrow \dots \xrightarrow{f_{(i-1)}} C_{(i)} \xrightarrow{f_{(i)}} \dots \xrightarrow{f_{(n-1)}} C_{(n)}, \tag{4.12}$$

by applying the homology construction to get the following diagram of vector spaces

$$0 \xrightarrow{\tilde{f}_{k,(0)}} H_{k,(1)} \xrightarrow{\tilde{f}_{k,(1)}} \dots \longrightarrow \dots \xrightarrow{\tilde{f}_{k,(i-1)}} H_{k,(i)} \xrightarrow{\tilde{f}_{k,(i)}} \dots \xrightarrow{\tilde{f}_{k,(n-1)}} H_{k,(n)}. \tag{4.13}$$

The persistent homology is a persistence module $\mathscr{M} = \{M_i, \varphi_i\}_{i=0}^{n}$ by setting $M_i = H_{k,(i)}$ and $\varphi_i = \tilde{f}_{k,(i)}$.

In the following section, we introduce a way of computing the persistent homology of a given sequence of chain complex connected by linear maps.

Construction of the persistent homology module

In this section, we present a construction of the k-persistent homology module $\{H_k^i, \varphi_i\}_{i\in\mathbb{N}}$ for general coefficients from a general chain complex $C = (C_\bullet, \partial_\bullet)$. For each index $i \in \mathbb{N}$, we first introduce alg. 4.8 for constructing the step H_k^i. Afterwards, we introduce alg. 4.9 for retrieving the structure map φ_i as the map at homology level induced by a general chain map $f : (C_\bullet, \partial_\bullet^C) \longrightarrow (D_\bullet, \partial_\bullet^D)$ in degree k. In the persistent homology module construction, f is one of chain maps $f_{(i)}$ connecting subsequent chain complexes.

Notice that these two constructions can be distributed and performed in parallel for each $i \in \mathbb{N}$.

Computing the homology steps in parallel

Let us introduce the following lemma:

Lemma 7 (Splitting [117]). *For a short exact sequence*

$$0 \longrightarrow K \xrightarrow{s} A \xrightarrow{q} Q \longrightarrow 0,$$

the following statements are equivalent:

a) *there exists a surjective map* $p : A \longrightarrow K$ *such that* $ps = id_K : K \longrightarrow K$

b) *there exists an injective map* $r : Q \longrightarrow A$ *such that* $qr = id_Q : Q \longrightarrow Q$

c) *there exists an isomorphism* $K \bigoplus Q \xrightarrow{h} A$ *such that* (id_K, h, id_Q) *is an isomorphisms of short exact sequences*

$$\begin{tikzcd} 0 \arrow[r] & K \arrow[r, "s"] & A \arrow[r, "q"] & Q \arrow[r] & 0 \\ 0 \arrow[r] & K \arrow[r, "incl_1"] \arrow[u, "id_K"] & K \oplus Q \arrow[r, "proj_2"] \arrow[u, "h"] & Q \arrow[r] \arrow[u, "id_Q"] & 0, \end{tikzcd} \tag{4.14}$$

where morphisms $incl_1$, $proj_2$ *are the canonical inclusion and projection of the direct sum* $\oplus$.

This is said that *the sequence splits*. Notice that in an Abelian category such as that of finite-dimensional vector spaces and linear maps, every exact sequence splits.

For computing the step H_k^i, we want to find a split sequence of type c of the Splitting Lemma above (lemma 7) for the short exact sequence given by $B_k \subseteq Z_k$ and by the definition of homology as $H_k^i = Z_k/B_k$

$$0 \longrightarrow B_k \hookrightarrow Z_k \xrightarrow{[\cdot]} H_k \longrightarrow 0,$$

where the arrow $\hookrightarrow$ denotes the natural injection and the map $[\cdot]$ is the natural quotient projection defining the k-homology space.

In particular, in diagram 4.14 the map h is the isomorphism assigning the standard basis of the biproduct to the basis $\{h_1, \ldots, h_{\beta_k}\}$ retrieved by Algorithm 4.8, and the maps incl_i and proj_i are the standard inclusions and projection of the component i in the biproduct space.

Algorithm 4.8 accepts as inputs the boundary matrices ∂_{k+1} and ∂_k expressed in terms of the standard basis of C. The algorithm reflects the standard idea introduced by Edelsbrunner et al ([108]) and called *left-to-right reduction*, and it boils down to computing the column reduction of the boundary matrices ∂_k and ∂_{k+1}. Let us call V_k and V_{k+1} the matrices such that $R_k = \partial_k V_k$ and $R_{k+1} = \partial_{k+1} V_{k+1}$ are matrices reduced by columns. The columns of V_k corresponding to zero columns of R_k provide the the basis $\{v_1,\dots,v_s\}$ for Z_k, whereas the non-zero columns of R_{k+1} provide the basis $\{b_1,\dots,b_r\}$ for B_k. Let us call $\beta_k = \dim H_k$. In the for-cycle, the left-to-right reduction is exploited again to obtain the result by completing the basis of B_k to a basis of Z_k using the vectors yielded by V_k.

Algorithm 4.8: Computing homology

Input: Boundary matrices $\partial_k, \partial_{k+1}$ of the chain complex C ;
Result: Betti number β_k and basis $\{h_1,\dots,h_{\beta_k},b_1,\dots,b_r\}$ of Z_k, where
    span$\{[h_1],\dots,[h_{\beta_k}]\} = H_k$ and span$\{b_1,\dots,b_r\} = B_k$.

```
Compute the reduction R_k = ∂_k V_k ;
Compute the reduction R_{k+1} = ∂_{k+1} V_{k+1} ;
b_1,...,b_r := non-zero columns of R_{k+1} ;
v_1,...,v_s := columns of V_k corresponding to zero columns of R_k ;
J := matrix with columns {b_1,...,b_r,v_1,...,v_s} ;
β_k = 1 ;
for i = r+1,...r+s do
    while ∃j < i s.t. low(J[i]) = low(J[i]) do
        l := low(J[i]);
        γ := J[l,i]/J[l,j];
        J[i] = J[i] − γJ[j];
    end
    if J[i] is non-zero then
        h_{β_k} := J[i];
        β_k = β_k + 1;
    end
end
return β_k, basis {h_1,...,h_{β_k},b_1,...,b_r}
```

Computing the homology structure maps in parallel

For any index i, Algorithm 4.9 computes the structure map φ_i as the map $\tilde{f}_k$ at homology level induced by a general linear map as the degree k of chain map $f_k : C_k \longrightarrow D_k$. Our algorithm assumes that the homology splittings $Z_k(C) \cong H_k(C) \oplus B_k(C)$ and $Z_k(D) \cong H_k(D) \oplus B_k(D)$ are already obtained by applying Algorithm 4.8.

In particular, this means that we have the splitting bases: $\{[h_1^C],\dots,[h_{\beta_k^C}^C]\}$ for $H_k(C)$, $\{[h_1^D],\dots,[h_{\beta_k^D}^D]\}$ for $H_k(D)$, and $\{b_1^D,\dots,b_r^D\}$ for $B_k(D)$.

Consider the map f_k as the degree k of a chain map f. By functoriality of homology, we have the following commutative diagram

$$
\begin{tikzcd}
0 \arrow[r] & B_k \arrow[r, hook] \arrow[d, dashed, "f|_k"] & Z_k \arrow[r, "{[\cdot]}"] \arrow[d, "f_k"] & H_k \arrow[r] \arrow[d, dashed, "\tilde{f}_k"] & 0 \\
0 \arrow[r] & B'_k \arrow[r, hook] & Z'_k \arrow[r, "{[\cdot]'}"] & H'_k \arrow[r] & 0,
\end{tikzcd}
\tag{4.15}
$$

where map $f_|$ is obvious by the fact that boundaries are mapped to boundaries.

Indeed, functoriality ofhomology depends on the map f_k preserving k-cycles and k-boundaries.
In particular for the map $\tilde{f}$ between quotient spaces, a choice of homology representative corresponds to a splitting map $r_k : H_k \longrightarrow Z_k$ satisfying case b) in Lemma 7.

This gives us a way of retrieving the map $\tilde{f}_k$ as

$$
\tilde{f}_k = [\cdot]' f_k r_k \tag{4.16}
$$

for some choice r_k of homology reprentatives, i.e., such that $[\cdot]r_k = \mathrm{id}_{H_k}$.

Theorem 9. *The map $\tilde{f}_k$ defined in alg. 4.9 is well-defined and it is the map induced by f_k through the homology functor.*

Proof. The map $\tilde{f}_k$ is obtained as $\tilde{f}_k = \mathrm{proj}_1 h^{-1} f_k h \mathrm{incl}_1$. Precomposing by $f_k h \mathrm{incl}_1$ implies applying f_k to $[h_i^C] \in H_k(C)$. The following composition of $\mathrm{proj}_1 h^{-1}$ implies that the image of $[h_i^C] \in H_k(C)$ under $\tilde{f}$ is independently retrieved by finding $\lambda_1,\dots,\lambda_{\beta_k^D}$ solving the linear system in the for-cycle of Algorithm 4.9.

This yields the desired map in the form $\tilde{f}_k([h_i^C]) = \sum_{j=1}^{\beta_k^D} \lambda_j(i)[h_j^D]$. □

Algorithm 4.9: Induced map between homology spaces

Input: Chain map $f_k : C_k \to D_k$, representatives cycles $h_1^C, \dots, h_{\beta_k^C}^C$ of a basis of $H_k(C)$, β_k^D and $\{h_1^D, \dots, h_{\beta_k^D}^D, b_1^D, \dots, b_r^D\}$ output of alg. 4.8 for D;

Result: map $\tilde{f}_k : H_k(C) \to H_k(D)$ induced by f_k.

$\tilde{f}_k :=$ zero matrix $\beta_k^D \times \beta_k^C$;

for $i = 1, \dots, \beta_k^C$ **do**

  Solve $f_k(h_i^C) = \sum_{j=1}^{\beta_k^D} \lambda_j h_j^D + \sum_{l=1}^{r} \mu_l b_l$;

  $\tilde{f}_k[i] = (\lambda_1, \dots, \lambda_{\beta_k^D})^T$

end

return $\tilde{f}_k$

Constructing the persistent homology through harmonics

In this section, we present a construction of the k-persistent homology module $\{\mathcal{H}_k^i, \hat{f}_i\}_{i\in\mathbb{N}}$ for coefficients in $\mathbb{R}$ through the space of *k-harmonics*. We call the persistence module $\{\mathcal{H}_k^i, \hat{f}_i\}_{i\in\mathbb{N}}$ the *harmonic persistence module*.

After some preliminaries on the Hodge Laplacian operator, by means of the Hodge decomposition (theorem 10), for each index $i \in \mathbb{N}$, we retrieve the persistence module step of the harmonic persistence module. Afterwards, we introduce Algorithm 4.10 for retrieving the structure map $\hat{f}_i$ as the map, at k-harmonics level, ensuring the isomorphism of persistence modules between the k-persistent homology module $\{H_k^i, \varphi_i\}$ and the harmonic persistence module $\{\mathcal{H}_k^i, \hat{f}_i\}$, proven in Theorem 12. Notice that this construction, as it was for the case of general coefficients, can be distributed and performed in parallel for each $i \in \mathbb{N}$.

The Hodge Laplacian operator

Since $\mathbb{R}$ has characteristic 0, given a chain complex

$$\dots \xrightarrow{\partial_{k+2}} C_{k+1} \xrightarrow{\partial_{k+1}} C_k \xrightarrow{\partial_k} C_{k-1} \xrightarrow{\partial_{k-1}} \dots \xrightarrow{\partial_2} C_1 \xrightarrow{\partial_1} C_0 \xrightarrow{\partial_0} 0,$$

we fix an inner product $\langle \cdot, \cdot \rangle_k$ on each space of k-chains C_k so that each operator ∂_k has a well-defined adjoint operator ∂_k^*, i.e.,$\langle \partial_k(c), d \rangle_k = \langle c, \partial_k^*(d) \rangle_k$.

We consider the sequence C^*

$$\dots \xleftarrow{\partial^*_{k+2}} C_{k+1} \xleftarrow{\partial^*_{k+1}} C_k \xleftarrow{\partial^*_k} C_{k-1} \xleftarrow{\partial^*_{k-1}} \dots \xleftarrow{\partial^*_2} C_1 \xleftarrow{\partial^*_1} C_0 \xleftarrow{\partial^*_0} 0.$$

For $k \in \mathbb{N}$, the *Hodge Laplacian* in degree k (Laplacian, for short) is the linear operator on k-chains $L_k : C_k \longrightarrow C_k$ given by

$$L_k := \partial_{k+1}\partial^*_{k+1} + \partial^*_k \partial_k. \tag{4.17}$$

Constructing the harmonic step in parallel

For each index $i \in \mathbb{N}$, the step in the harmonic persistence module is given by the space of *k-harmonics*. The space of k-harmonics of a chain complex is the subspace of C_k

$$\mathcal{H}_k := \ker(L_k). \tag{4.18}$$

Remark 5. If $h \in \mathcal{H}_k$, then $h \in \ker(\partial_k)$ and $h \in \ker(\partial^*_{k+1})$.

Indeed, both $\partial^*_k\partial_k$ and $\partial_{k+1}\partial^*_{k+1}$ are positive semidefinite, as $\langle c, \partial^*_k\partial_k c\rangle_k = \langle \partial_k c, \partial_k c\rangle_k \geq 0$ and $\langle c, \partial_{k+1}\partial^*_{k+1} c\rangle_k = \langle \partial^*_{k+1} c, \partial^*_{k+1} c\rangle_k \geq 0$. Hence, L_k is positive semidefinite, and $L_k c = 0$ implies $\partial^*_k \partial_k c = 0$ and $\partial_{k+1}\partial^*_{k+1} c = 0$. Thus, $\langle \partial_k c, \partial_k c\rangle_k = 0$ and $\langle \partial^*_{k+1} c, \partial^*_{k+1} c\rangle_k = 0$.

Elements of $\ker(\partial^*_{k+1})$ are called *k-cocycles*. We refer to [118] for more details.

Before moving to the computation of the structure maps in the harmonic persistence module, we discuss some useful results relative to the single step $\mathcal{H}^i_k$.

First of all, the Hodge decomposition theorem states that:

Theorem 10. *For a chain complex C and for every natural k,*

$$C_k = \mathcal{H}_k \oplus Im\ (\partial_{k+1}) \oplus Im\ (\partial^*_k)$$

Moreover, this decomposition is orthogonal and $Z_k = \mathcal{H}_k \oplus Im\ (\partial_{k+1})$.

By remark 5 , we can consider the homology class $[h]$ of an element $h \in \mathcal{H}_k$. As per Hodge theory ([119, 118]), it holds

Theorem 11. *The linear map $\psi_k : \mathcal{H}_k \longrightarrow H_k$ defined by $\psi_k(h) = [h]$ is an isomorphism.*

Given the Hodge decomposition there is the orthogonal projection $\pi_k : Z_k \to \mathcal{H}_k$ that is equal to the identity on $\mathcal{H}_k$ and sends the elements of B_k to zero.

Lemma 8. *For any k-cycle $z \in Z_k$, it holds $[z] = [\pi_k(z)]$ in H_k.*

Proof. Any k-cycle z can be written uniquely as $h+b$, where $h \in \mathcal{H}_k$ and $b \in B_k$. Since π_k sends boundaries to zero and it is the identity on $\mathcal{H}_k$, it holds $z - \pi_k(z) = h+b-h = b$, and therefore z and $\pi_k(z)$ are in the same homology class. □

Remark 6. If one chooses an orthonormal basis for the k-harmonics $\mathcal{H}_k$ and represent it via the matrix V_k whose columns are the basis vectors, then the projection π_k is represented by the matrix V_k^T.

Computing the harmonic structure map in parallel

In order to define the persistent homology with respect to harmonic representatives, we focus on the behavior of the harmonic subspace under the action of chain maps. On the one hand, we need to remark the following.

Remark 7. A chain map $f : C \longrightarrow D$ does not restrict to a map between the harmonic subspaces $\mathcal{H}_k^C$ and $\mathcal{H}_k^D$.

Indeed, given an element $h \in \mathcal{H}_k^C$, the k-cycle $f(h)$ is not necessarily in $\mathcal{H}_k^D$. More precisely, $f(h)$ is necessarily a k-cycle but not necessarily a k-cocycle.

On the other hand, we can define a map $\hat{f}_K : \mathcal{H}_k^C \to \mathcal{H}_k^D$ in the following way. Consider the natural inclusion $i_k^C : \mathcal{H}_k^C \to C_k$ and the projection $\pi_k^D : D_k \to \mathcal{H}_k^D$. We define

$$\hat{f}_k = \pi_k^D f_k i_k^C. \tag{4.19}$$

Notice that the above definition corresponds to the approach used to construct the induced map in homology in eq. (4.16).

Algorithm 4.10 computes Equation (4.19) by applying remark 6, the domain C and the target D to represent maps i_k^C and π_k^D.

Consider now a sequence of chain complexes connected by linear maps as in the sequence (4.12). We get induced the following sequence of linear maps between k-harmonic spaces

$$0 \xrightarrow{\hat{f}_{k,(0)}} \mathcal{H}_{k,(1)} \xrightarrow{\hat{f}_{k,(1)}} \ldots \longrightarrow \ldots \xrightarrow{\hat{f}_{k,(i-1)}} \mathcal{H}_{k,(i)} \xrightarrow{\hat{f}_{k,(i)}} \ldots \xrightarrow{\hat{f}_{k,(n-1)}} \mathcal{H}_{k,(n)}.$$

Theorem 12. *The DAPM's* $\{H_{k,(i)}, \tilde{f}_{k,(i)}\}_{i=0}^{n}$ *and*
$\{\mathcal{H}_{k,(i)}, \hat{f}_{k,(i)}\}_{i=0}^{n}$ *are isomorphic as persistence modules with coefficients in* $\mathbb{R}$*,*

Proof. It is enough to show that, for any chain map $f : C \longrightarrow D$, the following diagram commutes

$$\begin{array}{ccc} \mathcal{H}_k^C & \xrightarrow{\hat{f}_k} & \mathcal{H}_k^D \\ \big\downarrow \psi_k^C & & \big\downarrow \psi_k^D \\ H_p^C & \xrightarrow{\tilde{f}_k} & H_k^D. \end{array}$$

That is, for any $h \in \mathcal{H}_k^C$, it must be $\tilde{f}_k(\psi_k^C(h)) = \psi_k^D(\hat{f}_k(h))$. By the definition of $\tilde{f}_k$ and $\hat{f}_k$ the equality becomes $[f_k(h)] = [\pi_k^D(f_k(h))]$. Since $f_k(h)$ belongs to Z_k^D, the statements follows from lemma 8.

□

Algorithm 4.10: Induced map between Laplacian kernels

Input: Chain map $f_k : C_k(C) \to C_k(D)$, V_C basis of $\ker(L_k(C_k(C)))$, V_D basis of $\ker(L_k(C_k(D)))$;
Result: Matrix Φ representing $\hat{f}_k : \mathcal{H}_k(C) \to \mathcal{H}_k(D)$
$\Phi = V_D^T \cdot f_k \cdot V_C$;
return (Φ_i)

Conclusions

We now recap the contributions in this chapter. We have defined the concept of an interval basis as a suitable choice of vectors within a persistence module that generate it in the "freest" sense. We provide several different algorithms for the computation of an interval basis. Firstly, we describe a general *parallel* algorithm, that works for any persistence module over any field (alg. 4.2). Next, we specialize to the case of persistence modules over $\mathbb{R}$, where the scalar product structure can be leveraged to make use of the singular value decomposition, obtaining a more efficient method (alg. 4.5). Next again, we observe that a similar type of argument can be equally applied to the general algorithm, hence allowing to collapse the matrices as the reduction proceeds, and obtaining what we called an *improved* general algorithm (alg. 4.6).

At this point, after the concept of interval basis has been thoroughly analyzed, we introduce the relationship between an interval basis and the generators that can be obtained via a graded reduction of a presentation matrix of the persistence module. We piece together previous, unrelated contributions to construct a (*serial*) pipeline that leads from the same input, a DAPM, to an interval basis. We further observe that, in addition to what was presented in the relevant paper ([10]), we can implement the method without the need for symbolic computations.

Interestingly, the proof of the correctness of alg. 4.2 can then be seen as a constructive, alternative proof of the structure theorem for graded modules.

We give an estimate of the computational complexity of both approaches, and conclude that the *improved* algorithm 4.6, when implemented exploiting parallelism, is on average advantageous with respect to the matrix reduction one.

Finally, we specialize the approach to the persistence modules that derive from a filtration of simplicial complexes, i.e. from persistent homology proper (alg. 4.8, alg. 4.9). We describe how to obtain an entirely parallel pipeline for the computation of persistent homology over the reals, by the Hodge isomorphism relating homology to harmonics. We show how to construct in parallel induced maps between homology spaces from suitably-constructed maps between harmonics (alg. 4.10), and this concludes a parallel pipeline leading from a map of chain complexes (such as a filtration of simplicial complexes, the standard input in most TDA problems) all the way to an interval basis, through which one can additionally retrieve the barcode without additional effort.

Chapter 5

Future Perspectives

In this chapter, we explore some future avenues that could be explored as an expansion of the topics presented.

The subject of homology representatives is a vast one. We have touched on some of the most employed approaches to the problem, although we foresee that a complete answer to the question of choosing representatives is unlikely to be answered in general.
Minimal homology bases are now treated theoretically in a rather exhaustive manner. It appears that the road is blocked towards the computation of minimal generators in dimension higher than one. New methods can hopefully be developed to accelerate the computation of the 1D case, although the inherently combinatorial nature of the problem suggests that not much hope should be cast towards obtaining a very efficient algorithm.
Another delicate issue concerning minimal homology representatives is their well-posedness. We have gone to some lengths in the present thesis to discuss under what circumstances we can expect the minimal scaffold to be unique. However, the issue of the stability of the computed representatives remains rather elusive: it is clear that a small perturbation in the input may bring about a radical difference in the minimal representatives. One possible approach to this problem, for the suggestion of which we thank the input from the Referees, is akin to a bootstrapping method. Specifically, if we have at our disposal a reasonable statistical model of the noise in our input complex, we can obtain multiple samples of our perturbed data, computing

minimal representatives several times around. This can yield an "average" set of fuzzy generators, which could potentially capture a robust picture of the topology of our data. It must however be noted that this approach increases considerably the complexity of the scaffolding pipeline.
The mixed approaches such as the homologically-persistent skeleton described here, on the other hand, are promising in terms of their applicability, due to a more reasonable computational cost. We foresee that a natural development of the work reported in the present thesis would be to find a suitable application (i.e. one satisfying the metricity requirement) where representatives of homology are of interest, such as in metric graphs, cristallography, material science, and so on. Further, the extent to which the analogy between minimal generators and approximately minimal ones holds could deserve further enquiry. For example, an interesting question concerns the effect of applying a computationally efficient complex reduction via one of the many algorithms for simplex collapse as a pre-processing step. Applying the computation of minimal representatives to this more manageable input could save a considerable amount of computation, but the significance of the obtained output would require further study.

Additionally, the topic of minimal (or rather, approximately minimal) generators has recently been attacked from an entirely novel point of view. It is a relatively classical topic to feed the invariants computed via persistent homology into a machine learning pipeline. In very recent years, however, schemes have emerged to *apply* machine learning directly to the topology of the dataset, using persistent homology as a guiding loss function instead of as a descriptor. Works that laid the ground on the subject are, for example, [120–123]. Within this context, work has started to emerge that attempts to address the topic of finding minimal generators of homology classes by leveraging topological information, usually encoded in the Hodge Laplacian operator. We suggest, as examples, the works [124, 125].

The section about using Alexander duality in the search for homology representatives is clearly the one where more work to be carried out is easily found. Most of the concepts presented in the section are ideas that need to be more carefully defined, and then formally proved.

The topic of interval bases relates to several other construction in the literature, and we foresee that it may deserve some work towards the definition of similar objects in multipersistent homology, where one is not given the strong guarantees of the structure theorem, due to the wilder nature of $\mathbb{F}[x_1,...,x_n]$-modules.
Another issue that we are currently investigating that relates to interval bases is that of how an interval basis can inform the construction of a matching between two barcodes, when the two corresponding persistence modules are linked by a morphism. The topic of induced matching has already attracted attention, resulting in (at least) the works [126–128]. Here we sketch the construction of such a matching.
The recent work [128] has detected the condition of not containing nested bars as necessary to induce a partial matching from a persistence module morphism in terms of barcode bases, that is a choice of a graded basis along the persistence module from which the barcode can be easily read off. Unfortunately, most applications deal with barcodes containing nested bars.
Rather than in terms of barcode bases, an equivalent way of describing an equioriented persistence module (which excludes the zig-zag modules described in the paper) was introduced in the previous chapter (based on [102]) in terms of an interval basis, that is a choice of generators such that the cyclic submodules generated by each generator directly provide the persistence module decomposition into interval modules. Based on it, we have proposed another way of inducing a mapping of barcodes when persistence module morphisms are expressed in terms of interval bases. By exploiting the first homomorphisms theorem for graded modules, we recover a third barcode to which both the source and the target barcodes can be mapped to and, crucially, to which generators can be tracked. This allows us to circumvent the limitation given by the nested bars, while still choosing a matching in a non-arbitrary sense, via certain *shrinking* operations that are defined in terms of the kernel of the morphism.
We have found that this map is computable by a modified version of the general algorithm 4.2 to obtain the interval basis of a given submodule.
The possibility of tracking generators along a given morphism of persistence modules might be of interest, for instance, in the case of multiparameter persistent homology, which contains multiple morphisms of one-parameter persistence modules.

References

[1] Saunders Mac Lane. *Categories for the Working Mathematician*, volume 5 of *Graduate Texts in Mathematics*. Springer New York, New York, NY, 1978.

[2] Brendan Fong and David I. Spivak. *An Invitation to Applied Category Theory: Seven Sketches in Compositionality*. Cambridge University Press, 7 2019.

[3] Emily Riehl. *Category Theory in Context*. Courier Dover Publications, 2016.

[4] Thomas Scott Blyth. *Module theory : an approach to linear algebra*. Oxford University Press, 1990.

[5] Heather A. Harrington, Nina Otter, Hal Schenck, and Ulrike Tillmann. Stratifying Multiparameter Persistent Homology. *https://doi.org/10.1137/18M1224350*, 3(3):439–471, 9 2019.

[6] Maximilian Neumann. Multidimensional Persistence: Invariants and Parameterization. *ArXiv*, 8 2021.

[7] Cary Webb. Decomposition of graded modules. *Proceedings of the American Mathematical Society*, 94(4):565–571, 1985.

[8] A. J. Zomorodian and G. Carlsson. Computing persistent homology. *Discrete & Computational Geometry*, 33(2):249–274, 2005.

[9] David Eisenbud. Commutative Algebra. 150, 1995.

[10] Primoz Skraba and Mikael Vejdemo-Johansson. Persistence modules: Algebra and algorithms. *MATHEMATICS OF COMPUTATION*, 00(0), 2 2013.

[11] H. Edelsbrunner and J. Harer. Computational topology: An introduction. *American Mathematical Society*, 2010.

[12] René Corbet and Michael Kerber. The representation theorem of persistence revisited and generalized. *Journal of Applied and Computational Topology*, 2(1-2):1–31, 10 2018.

[13] Peter Bubenik and Jonathan A. Scott. Categorification of Persistent Homology. *Discrete & Computational Geometry 2014 51:3*, 51(3):600–627, 1 2014.

[14] Magnus Bakke Botnan and William Crawley-Boevey. Decomposition of persistence modules. *Proceedings of the American Mathematical Society*, 148(11):4581–4596, 8 2020.

[15] Chao Chen and Daniel Freedman. Hardness results for homology localization. *Discrete & Computational Geometry*, 45(3):425–448, Apr 2011.

[16] Mark EJ Newman. The structure and function of complex networks. *SIAM review*, 45(2):167–256, 2003.

[17] Alain Barrat, Marc Barthelemy, Romualdo Pastor-Satorras, and Alessandro Vespignani. The architecture of complex weighted networks. *Proceedings of the national academy of sciences*, 101(11):3747–3752, 2004.

[18] Mark S. Granovetter. The Strength of Weak Ties. *American Journal of Sociology*, 78(6):1360–1380, 1973.

[19] Fernando Vega-Redondo. Complex social networks. *Cambridge University Press*, 2007.

[20] Romualdo Pastor-Satorras, Claudio Castellano, Piet Van Mieghem, and Alessandro Vespignani. Epidemic processes in complex networks. *Reviews of modern physics*, 87(3):925, 2015.

[21] Vittoria Colizza, Alain Barrat, Marc Barthélemy, and Alessandro Vespignani. The role of the airline transportation network in the prediction and predictability of global epidemics. *Proceedings of the National Academy of Sciences*, 103(7):2015–2020, 2006.

[22] Michelle Girvan and Mark EJ Newman. Community structure in social and biological networks. *Proceedings of the national academy of sciences*, 99(12):7821–7826, 2002.

[23] Uri Alon. Biological networks: the tinkerer as an engineer. *Science*, 301(5641):1866–1867, 2003.

[24] Danielle S Bassett and Olaf Sporns. Network neuroscience. *Nature neuroscience*, 20(3):353, 2017.

[25] Ed Bullmore and Olaf Sporns. Complex brain networks: graph theoretical analysis of structural and functional systems. *Nature reviews neuroscience*, 10(3):186–198, 2009.

[26] Danielle Smith Bassett and ED Bullmore. Small-world brain networks. *The neuroscientist*, 12(6):512–523, 2006.

[27] Danijela Horak, Slobodan Maletić, and Milan Rajković. Persistent homology of complex networks. *Journal of Statistical Mechanics: Theory and Experiment*, 2009(03):P03034, mar 2009.

[28] Alice Patania, Giovanni Petri, and Francesco Vaccarino. The shape of collaborations. *EPJ Data Science*, 6(1):18, Aug 2017.

[29] Hyekyoung Lee, Moo K Chung, Hyejin Kang, Bung-Nyun Kim, and Dong Soo Lee. Discriminative persistent homology of brain networks. In *2011 IEEE International Symposium on Biomedical Imaging: From Nano to Macro*, pages 841–844. IEEE, 2011.

[30] B. Rieck, U. Fugacci, J. Lukasczyk, and H. Leitte. Clique community persistence: A topological visual analysis approach for complex networks. *IEEE Transactions on Visualization and Computer Graphics*, 24(1):822–831, Jan 2018.

[31] Yasuaki Hiraoka, Takenobu Nakamura, Akihiko Hirata, Emerson G. Escolar, Kaname Matsue, and Yasumasa Nishiura. Hierarchical structures of amorphous solids characterized by persistent homology. *Proceedings of the National Academy of Sciences of the United States of America*, 113(26):7035–7040, 2016.

[32] Yongjin Lee, Senja D Barthel, Paweł Dłotko, S Mohamad Moosavi, Kathryn Hess, and Berend Smit. Quantifying similarity of pore-geometry in nanoporous materials. *Nature communications*, 8(1):1–8, 2017.

[33] Joseph Minhow Chan, Gunnar Carlsson, and Raul Rabadan. Topology of viral evolution. *Proceedings of the National Academy of Sciences*, 110(46):18566–18571, 2013.

[34] Zhenyu Meng, D. Vijay Anand, Yunpeng Lu, Jie Wu, and Kelin Xia. Weighted persistent homology for biomolecular data analysis. *Scientific Reports*, 10, 2020.

[35] M. K. Chung, P. Bubenik, and P. T. Kim. Persistence diagrams of cortical surface data. In *Information Processing in Medical Imaging*, pages 386–397. Springer, 2009.

[36] M.-L. Dequeant, S. Ahnert, H. Edelsbrunner, T. Fink, E. Glynn, G. Hattem, A. Kudlicki, Y. Mileyko, J. Morton, A. Mushegian, et al. Comparison of pattern detection methods in microarray time series of the segmentation clock. *PLoS One*, 3(8):e2856, 2008.

[37] Y. Wang, P. K. Agarwal, P. Brown, Edelsbrunner H, and J. Rudolph. Coarse and reliable geometric alignment for protein docking. In *In Proceedings of Pacific Symposium on Biocomputing*, volume 10, pages 65–75, 2005.

[38] S. Martin, A. Thompson, E. A. Coutsias, and J.-P. Watson. Topology of cyclooctane energy landscape. *Journal of Chemical Physics*, 132(23):234115, 2010.

[39] Angkoon Phinyomark, Rami N Khushaba, Esther Ibáñez-Marcelo, Alice Patania, Erik Scheme, and Giovanni Petri. Navigating features: a topologically informed chart of electromyographic features space. *Journal of The Royal Society Interface*, 14(137):20170734, 2017.

[40] Vin De Silva and Robert Ghrist. Coverage in sensor networks via persistent homology. *Algebraic & Geometric Topology*, 7(1):339–358, 2007.

[41] Rien van de Weygaert, Gert Vegter, Herbert Edelsbrunner, Bernard J. T. Jones, Pratyush Pranav, Changbom Park, Wojciech A. Hellwing, Bob Eldering, Nico Kruithof, E. G. P. (Patrick) Bos, and et al. *Alpha, Betti and the Megaparsec Universe: On the Topology of the Cosmic Web*, page 60–101. Springer-Verlag, Berlin, Heidelberg, 2011.

[42] Alice Patania, Pierluigi Selvaggi, Mattia Veronese, Ottavia Dipasquale, Paul Expert, and Giovanni Petri. Topological gene expression networks recapitulate brain anatomy and function. *Network Neuroscience*, 3(3):744–762, 2019.

[43] Peter Lawson, Andrew B Sholl, J Quincy Brown, Brittany Terese Fasy, and Carola Wenk. persistent homology for the quantitative evaluation of architectural features in prostate cancer histology. *Scientific reports*, 9(1):1–15, 2019.

[44] Chad Giusti, Eva Pastalkova, Carina Curto, and Vladimir Itskov. Clique topology reveals intrinsic geometric structure in neural correlations. *Proceedings of the National Academy of Sciences of the United States of America*, 112(44):13455–13460, 2015.

[45] Yuan Wang, Hernando Ombao, and Moo K. Chung. Topological data analysis of single-trial electroencephalographic signals. *Annals of Applied Statistics*, 12(3):1506–1534, 2017.

[46] Jaejun Yoo, Eun Young Kim, Yong Min Ahn, and Jong Chul Ye. Topological persistence vineyard for dynamic functional brain connectivity during resting and gaming stages. *Journal of Neuroscience Methods*, 267(15):1–13, 2016.

[47] G. Petri, P. Expert, F. Turkheimer, R. Carhart-Harris, D. Nutt, P. J. Hellyer, and F. Vaccarino. Homological scaffolds of brain functional networks. *Journal of The Royal Society Interface*, 11(101):20140873, 2014.

[48] Esther Ibáñez-Marcelo, Lisa Campioni, Angkoon Phinyomark, Giovanni Petri, and Enrica L Santarcangelo. Topology highlights mesoscopic functional equivalence between imagery and perception: The case of hypnotizability. *NeuroImage*, 200:437–449, 2019.

[49] Louis-David Lord, Paul Expert, Henrique M Fernandes, Giovanni Petri, Tim J Van Hartevelt, Francesco Vaccarino, Gustavo Deco, Federico Turkheimer, and Morten L Kringelbach. Insights into brain architectures from the homological scaffolds of functional connectivity networks. *Frontiers in systems neuroscience*, 10:85, 2016.

[50] Esther Ibáñez-Marcelo, Lisa Campioni, Diego Manzoni, Enrica L Santarcangelo, and Giovanni Petri. Spectral and topological analyses of the cortical representation of the head position: Does hypnotizability matter? *Brain and behavior*, 9(6):e01277, 2019.

[51] Wei Guo and Ashis G. Banerjee. Toward automated prediction of manufacturing productivity based on feature selection using topological data analysis. In *IEEE International Symposium on Assembly and Manufacturing*, pages 31–36, 2016.

[52] Angkoon Phinyomark, Giovanni Petri, Esther Ibáñez-Marcelo, Sean T Osis, and Reed Ferber. Analysis of big data in gait biomechanics: Current trends and future directions. *Journal of medical and biological engineering*, 38(2):244–260, 2018.

[53] Evan Campbell, Angkoon Phinyomark, Ali H Al-Timemy, Rami N Khushaba, Giovanni Petri, and Erik Scheme. Differences in emg feature space between able-bodied and amputee subjects for myoelectric control. In *2019 9th International IEEE/EMBS Conference on Neural Engineering (NER)*, pages 33–36. IEEE, 2019.

[54] Alice Patania, Giovanni Petri, and Francesco Vaccarino. The shape of collaborations. *EPJ Data Science*, 6(1):18, 2017.

[55] Austin R Benson, Rediet Abebe, Michael T Schaub, Ali Jadbabaie, and Jon Kleinberg. Simplicial closure and higher-order link prediction. *Proceedings of the National Academy of Sciences*, 115(48):E11221–E11230, 2018.

[56] A. Verri, C. Uras, P. Frosini, and M. Ferri. On the use of size functions for shape analysis. *Biological Cybernetics 1993 70:2*, 70(2):99–107, 12 1993.

[57] Massimo Ferri, Sandra Lombardini, and Clemente Pallotti. Leukocyte classification by size functions. *IEEE Workshop on Applications of Computer Vision - Proceedings*, pages 223–229, 1994.

[58] Claudio Uras and Alessandro Verri. Computing Size Functions from Edge Maps. *International Journal of Computer Vision 1997 23:2*, 23(2):169–183, 1997.

[59] Ginestra Bianconi and Robert M. Ziff. Topological percolation on hyperbolic simplicial complexes. *Phys. Rev. E*, 98:052308, Nov 2018.

[60] Ana P Millán, Joaquín J Torres, and Ginestra Bianconi. Explosive higher-order kuramoto dynamics on simplicial complexes. *Phys. Rev. Lett.*, page to appear, 2019.

[61] Harish Kannan, Emil Saucan, Indrava Roy, and Areejit Samal. Persistent homology of unweighted complex networks via discrete morse theory. *Scientific Reports*, 9, 2019.

[62] Giovanni Petri, Martina Scolamiero, Irene Donato, and Francesco Vaccarino. Topological strata of weighted complex networks. *PloS one*, 8(6), 2013.

[63] Alice Patania, Francesco Vaccarino, and Giovanni Petri. Topological analysis of data. *EPJ Data Science*, 6(1):7, 2017.

[64] Irene Donato, Matteo Gori, Marco Pettini, Giovanni Petri, Sarah De Nigris, Roberto Franzosi, and Francesco Vaccarino. Persistent homology analysis of phase transitions. *Physical Review E*, 93(5):052138, 2016.

[65] Ann Sizemore, Chad Giusti, and Danielle S Bassett. Classification of weighted networks through mesoscale homological features. *Journal of Complex Networks*, 5(2):245–273, 2017.

[66] V. Kurlin. A one-dimensional homologically persistent skeleton of an unstructured point cloud in any metric space. *Computer Graphics Forum*, 34(5):253–262, 2015.

[67] S. Kalisnik, V. Kurlin, and D. Lesnik. A higher-dimensional homologically persistent skeleton. *Advances in Applied Mathematics*, 102:113–142, 2019.

[68] Xiaoyin Ge, Issam I. Safa, Mikhail Belkin, and Yusu Wang. Data skeletonization via Reeb graphs. *Advances in Neural Information Processing Systems 24*, pages 837–845, 2011.

[69] Frédéric Chazal, Ruqi Huang, and Jian Sun. Gromov–hausdorff approximation of filamentary structures using reeb-type graphs. *Discrete & Computational Geometry*, 53(3):621–649, 2015.

[70] Ann E Sizemore, Chad Giusti, Ari Kahn, Jean M Vettel, Richard F Betzel, and Danielle S Bassett. Cliques and cavities in the human connectome. *Journal of computational neuroscience*, 44(1):115–145, 2018.

[71] Ippei Obayashi. Volume-optimal cycle: Tightest representative cycle of a generator in persistent homology. *SIAM Journal on Applied Algebra and Geometry*, 2(4):508–534, 2018.

[72] Tamal Dey, Jian Sun, and Yusu Wang. Approximating loops in a shortest homology basis from point data. *Proceedings of the Annual Symposium on Computational Geometry*, 09 2009.

[73] Tamal K Dey, Tianqi Li, and Yusu Wang. Efficient algorithms for computing a minimal homology basis. In *Latin American Symposium on Theoretical Informatics*, pages 376–398. Springer, 2018.

[74] Andrea Baronchelli, Ramon Ferrer-i Cancho, Romualdo Pastor-Satorras, Nick Chater, and Morten H Christiansen. Networks in cognitive science. *Trends in cognitive sciences*, 17(7):348–360, 2013.

[75] Pek Y Lum, Gurjeet Singh, Alan Lehman, Tigran Ishkanov, Mikael Vejdemo-Johansson, Muthu Alagappan, John Carlsson, and Gunnar Carlsson. Extracting insights from the shape of complex data using topology. *Scientific reports*, 3:1236, 2013.

[76] Andrew Tausz, Mikael Vejdemo-Johansson, and Henry Adams. JavaPlex: A research software package for persistent (co)homology. In Han Hong and Chee Yap, editors, *Proceedings of ICMS 2014*, Lecture Notes in Computer Science 8592, pages 129–136, 2014.

[77] Cecil Jose A. Delfinado and Herbert Edelsbrunner. An incremental algorithm for betti numbers of simplicial complexes on the 3-sphere. *Computer Aided Geometric Design*, 12(7):771 – 784, 1995. Grid Generation, Finite Elements, and Geometric Design.

[78] J. Horton. A polynomial-time algorithm to find the shortest cycle basis of a graph. *SIAM Journal on Computing*, 16(2):358–366, 1987.

[79] J. C. de Pina. Applications of shortest path methods. *PhD Thesis University of Amsterdam*, 1, 1995.

[80] Telikepalli Kavitha, Kurt Mehlhorn, Dimitrios Michail, and Katarzyna Paluch. A faster algorithm for minimum cycle basis of graphs. In Josep Díaz, Juhani Karhumäki, Arto Lepistö, and Donald Sannella, editors, *Automata, Languages and Programming*, pages 846–857, Berlin, Heidelberg, 2004. Springer Berlin Heidelberg.

[81] Oleksiy Busaryev, Sergio Cabello, Chao Chen, Tamal K Dey, and Yusu Wang. Annotating simplices with a homology basis and its applications. In *Scandinavian workshop on algorithm theory*, pages 189–200. Springer, 2012.

[82] Don Coppersmith and Shmuel Winograd. Matrix multiplication via arithmetic progressions. *Journal of Symbolic Computation*, 9(3):251 – 280, 1990. Computational algebraic complexity editorial.

[83] François Le Gall. Powers of tensors and fast matrix multiplication. In *Proceedings of the 39th International Symposium on Symbolic and Algebraic Computation*, ISSAC '14, pages 296–303, New York, NY, USA, 2014. ACM.

[84] Marco Guerra and Alessandro De Gregorio. GitHub Repository MinScaffold, 2019.

[85] Ulrich Bauer. Ripser: efficient computation of Vietoris–Rips persistence barcodes. *Journal of Applied and Computational Topology*, 5(3):391–423, 9 2021.

[86] A. Jaillard M. Termenon, C. Delon-Martin and S. Achard. Reliability of graph analysis of resting state fmri using test-retest dataset from the human connectome project. *Neuroimage*, 142(15):172–187, 2016.

[87] Alexandre Abraham, Fabian Pedregosa, Michael Eickenberg, Philippe Gervais, Andreas Mueller, Jean Kossaifi, Alexandre Gramfort, Bertrand Thirion, and Gael Varoquaux. Machine learning for neuroimaging with scikit-learn. *Frontiers in Neuroinformatics*, 8:14, 2014.

[88] Rossana Mastrandrea, Andrea Gabrielli, Fabrizio Piras, Gianfranco Spalletta, Guido Caldarelli, and Tommaso Gili. Organization and hierarchy of the human functional brain network lead to a chain-like core. *Scientific Reports*, 7(1):1–13, 2017.

[89] Philip I. Davies and Nicholas J. Higham. Numerically stable generation of correlation matrices and their factors. *BIT Numerical Mathematics*, 40(4):640–651, 2000.

[90] Anna Katharina Kuhlen, Carsten Allefeld, and John-Dylan Haynes. Content-specific coordination of listeners' to speakers' eeg during communication. *Frontiers in human neuroscience*, 6:266, 2012.

[91] Bosiljka Tadić, Miroslav Andjelković, and Milovan Šuvakov. Origin of hyperbolicity in brain-to-brain coordination networks. *Frontiers in Physics*, 6:7, 2018.

[92] Maïté Termenon, Assia Jaillard, Chantal Delon-Martin, and Sophie Achard. Reliability of graph analysis of resting state fmri using test-retest dataset from the human connectome project. *Neuroimage*, 142:172–187, 2016.

[93] Miroslav Andjelković, Bosiljka Tadić, and Roderick Melnik. The topology of higher-order complexes associated with brain hubs in human connectomes. *Scientific reports*, 10(1):1–10, 2020.

[94] Alessandro Muscoloni and Carlo Vittorio Cannistraci. A nonuniform popularity-similarity optimization (npso) model to efficiently generate realistic complex networks with communities. *New Journal of Physics*, 20(5):052002, 2018.

[95] Muscoloni Alessandro and Cannistraci Carlo Vittorio. Leveraging the nonuniform pso network model as a benchmark for performance evaluation in community detection and link prediction. *New Journal of Physics*, 20(6):063022, 2018.

[96] Benjamin Schweinhart. *Statistical Topology of Embedded Graphs*. PhD thesis, Princeton University, 2015.

[97] Afra Zomorodian and Gunnar Carlsson. Localized homology. *Computational Geometry*, 41(3):126–148, 11 2008.

[98] Tamal K. Dey, Anil N. Hirani, and Bala Krishnamoorthy. Optimal Homologous Cycles, Total Unimodularity, and Linear Programming. *https://doi.org/10.1137/100800245*, 40(4):1026–1044, 7 2011.

[99] Lu Li, Connor Thompson, Gregory Henselman-Petrusek, Chad Giusti, and Lori Ziegelmeier. Minimal Cycle Representatives in Persistent Homology using Linear Programming: an Empirical Study with User's Guide. *ArXiv*, 5 2021.

[100] Haibin Hang, Chad Giusti, Lori Ziegelmeier, and Gregory Henselman-Petrusek. U-match factorization: sparse homological algebra, lazy cycle representatives, and dualities in persistent (co)homology. *ArXiv*, 8 2021.

[101] James R. Munkres. *Elements of algebraic topology*. CRC Press, 1 2018.

[102] Alessandro De Gregorio, Marco Guerra, Sara Scaramuccia, and Francesco Vaccarino. Parallel decomposition of persistence modules through interval bases. *ArXiv*, 6 2021.

[103] H. Edelsbrunner, D. Letscher, and A. Zomorodian. Topological persistence and simplification. *Discrete & Computational Geometry*, 28(4):511–533, Nov 2002.

[104] Stefania Ebli and Gard Spreemann. A notion of harmonic clustering in simplicial complexes. In *Proceedings - 18th IEEE International Conference on Machine Learning and Applications, ICMLA 2019*, pages 1083–1090. Institute of Electrical and Electronics Engineers Inc., 12 2019.

[105] Rui Wang, Duc Duy Nguyen, and Guo-Wei Wei. Persistent spectral graph. *International Journal for Numerical Methods in Biomedical Engineering*, 36(9):e3376, 2020.

[106] Rui Wang, Rundong Zhao, Emily Ribando-Gros, Jiahui Chen, Yiying Tong, and Guo-Wei Wei. HERMES: Persistent spectral graph software. *Foundations of Data Science*, 3(1):67, 3 2021.

[107] Herbert Edelsbrunner, David Letscher, and Afra Zomorodian. Topological persistence and simplification. *Discrete and Computational Geometry*, 28(4):511–533, 11 2002.

[108] Cecil Jose A. Delfinado and Herbert Edelsbrunner. An incremental algorithm for Betti numbers of simplicial complexes on the 3-sphere. *Computer Aided Geometric Design*, 12(7):771–784, 11 1995.

[109] Gunnar Carlsson and Afra Zomorodian. Computing Persistent Homology. *Discrete & Computational Geometry*, 33(2):249–274, 2005.

[110] Marco Guerra, Alessandro De Gregorio, Ulderico Fugacci, Giovanni Petri, and Francesco Vaccarino. Homological scaffold via minimal homology bases. *Scientific reports*, 11(1):5355, 3 2021.

[111] Ulrich Bauer, Michael Kerber, and Jan Reininghaus. Clear and compress: Computing persistent homology in chunks. *Mathematics and Visualization*, 0(9783319040981):103–117, 2014.

[112] Tamal K. Dey, Fengtao Fan, and Yusu Wang. Computing topological persistence for simplicial maps. In *Proceedings of the Annual Symposium on Computational Geometry*, pages 345–354. Association for Computing Machinery, 8 2014.

[113] Michael Kerber and Hannah Schreiber. Barcodes of Towers and a Streaming Algorithm for Persistent Homology. *Discrete and Computational Geometry*, 61(4):852–879, 6 2019.

[114] Gunnar Carlsson and Vin de Silva. Zigzag Persistence. *Foundations of Computational Mathematics*, 10(4):367–405, 8 2010.

[115] Gunnar Carlsson, Anjan Dwaraknath, and Bradley J. Nelson. Persistent and Zigzag Homology: A Matrix Factorization Viewpoint. *arXiv*, 11 2019.

[116] Frédéric Chazal, Vin de Silva, Marc Glisse, and Steve Oudot. The Structure and Stability of Persistence Modules. *Springer*, 2016.

[117] Allen Hatcher. *Algebraic topology*. Cambridge Univ. Press, 2000.

[118] Danijela Horak and Jürgen Jost. Spectra of combinatorial Laplace operators on simplicial complexes. *Advances in Mathematics*, 244:303–336, 2013.

[119] Beno Eckmann. Harmonische Funktionen und Randwertaufgaben in einem Komplex. *Commentarii Mathematici Helvetici*, 17(1):240–255, 12 1944.

[120] Rickard Brüel-Gabrielsson, Bradley J. Nelson, Anjan Dwaraknath, Primoz Skraba, Leonidas J. Guibas, and Gunnar Carlsson. A topology layer for machine learning. In *Proceedings of the 23rdInternational Conference on Artificial Intelligence and Statistics (AISTATS) 2020*, volume 108, 2020.

[121] Mathieu Carrière, Frédéric Chazal, Marc Glisse, Yuichi Ike, and Hariprasad Kannan. Optimizing persistent homology based functions. *ArXiv*, 2021.

[122] Michael Moor, Max Horn, Bastian Rieck, and Karsten Borgwardt. Topological Autoencoders. In *Proceedings of the 37th International Conference on Machine Learning*, Proceedings of Machine Learning Research, 119, Vienna, Austria, 2020.

[123] Jacob Leygonie, Steve Oudot, and Ulrike Tillmann. A Framework for Differential Calculus on Persistence Barcodes. *arXiv*, 10 2019.

[124] Yu-Chia Chen and Marina Meil˘. The decomposition of the higher-order homology embedding constructed from the k-Laplacian. *ArXiv*, 2021.

[125] Alexandros Dimitrios Keros, Vidit Nanda, and Kartic Subr. Dist2Cycle: A Simplicial Neural Network for Homology Localization. *ArXiv*, 10 2021.

[126] Ulrich Bauer and Michael Lesnick. Induced Matchings and the Algebraic Stability of Persistence Barcodes. *Journal of Computational Geometry*, 6, 11 2013.

[127] R. Gonzalez-Diaz, M. Soriano-Trigueros, and A. Torras-Casas. Partial Matchings Induced by Morphisms between Persistence Modules. *ArXiv*, 7 2021.

[128] Emile Jacquard, Vidit Nanda, and Ulrike Tillmann. The Space of Barcode Bases for Persistence Modules. *ArXiv*, 11 2021.