ETH
Eidgenössische Technische Hochschule Zürich
Swiss Federal Institute of Technology Zurich

# Persistent Homology and Classification of Neuronal Morphologies

Bachelor Thesis

Melissa Daniele
medaniele@student.ethz.ch

17.07.2024

Supervisor: Dr. Sara Kališnik Hintz

Department of Mathematics, ETH Zürich

## Abstract

At the core of the analysis of the functionality and structure of the brain is the research of its fundamental units, the neurons. In particular, the shape of the neuron has proven to be the most influential factor in understanding the functionality of the brain. For this reason, scientists began to observe neurons under the microscope and classify them by properties, such as their branching structure or total length. This created the first classifications of neurons by their morphology. In this thesis, we introduce a method for the classification of neuronal morphologies, called the *Topological Morphology Descriptor* (TMD). This descriptor captures the shape of the neuron by associating to it a multiset of intervals, called a *barcode*. We also present a stochastic right inverse to the TMD algorithm, the *Topological Neuronal Synthesis* (TNS), that synthesizes a neuron from a given barcode. With the goal of developing these methods, we first introduce *trees*, *simplicial complexes* and *homology*. The former is one of the most significant invariants of *algebraic topology* measuring the number of holes, voids and higher-dimensional cavities. We then extend the notion of homology to the setting of filtered spaces. This extension gives rise to the notion of *persistent vector spaces* and enables the construction of an invariant, similar to homology, that captures the shape of finite metric spaces. This invariant, on which both the TMD and TNS are based, is called *persistent homology*.

## Acknowledgements

I would like to express my biggest gratitude to Dr. Sara Kališnik Hintz, who introduced me to the world of Topological Data Analysis and gave me the opportunity to conduct a deeper study in such a fascinating and intriguing branch of mathematics. Her guidance was invaluable, and she was always present whenever I needed help understanding mathematical concepts or organizing the structure of the thesis.

I am also deeply grateful to Indi, who took his time to read my thesis thoroughly and supported me whenever I needed encouragement. There are no words to express my thankfulness for his presence.

Many thanks also to Eva Lia, for the constant support and kindness she has offered me, not only during the writing of this thesis, but throughout all the time we've known each other.

To mamma, papà e Pietro, thank you for always believing in me and doing everything to help me achieve my dreams. Without you, none of this would have been possible.

# Contents

**Contents** **iii**

**1 Introduction** **1**

**2 Trees, Simplicial Complexes and Homology** **7**
- 2.1 Graphs . . . . . . . . . . 7
- 2.2 Trees . . . . . . . . . . 11
  - 2.2.1 Combinatorial Trees . . . . . . . . . . 12
  - 2.2.2 Merge Trees . . . . . . . . . . 13
- 2.3 Simplicial Complexes . . . . . . . . . . 15
- 2.4 Homology . . . . . . . . . . 19

**3 Filtrations and Persistent Homology** **25**
- 3.1 Filtrations . . . . . . . . . . 25
- 3.2 Persistent Vector Spaces . . . . . . . . . . 27
- 3.3 Decomposition Theorem of Persistent Vector Spaces . . . . . . 32
  - 3.3.1 Persistent Homology Computation . . . . . . . . . . 36
  - 3.3.2 Persistent Diagrams and Barcodes . . . . . . . . . . 41
- 3.4 Bottleneck Distance . . . . . . . . . . 42

**4 Neuronal Morphology Characterisation** **47**
- 4.1 Neurons . . . . . . . . . . 47
- 4.2 From Trees to Barcodes . . . . . . . . . . 49
  - 4.2.1 Elder Rule . . . . . . . . . . 49
  - 4.2.2 Strict Barcodes . . . . . . . . . . 50
  - 4.2.3 Topological Morphology Descriptor . . . . . . . . . . 53
- 4.3 Topological Neuronal Synthesis . . . . . . . . . . 55
  - 4.3.1 Bifurcation and Termination Process . . . . . . . . . . 56
  - 4.3.2 Elongation . . . . . . . . . . 57
- 4.4 Tree Realization Number . . . . . . . . . . 59

4.4.1 Alteration of Barcodes . . . . . . . . . . . . . . . . . . . . 61
4.5 TMD $\circ$ TNS . . . . . . . . . . . . . . . . . . . . . . . . . . . 63
4.5.1 Similarity of Barcodes . . . . . . . . . . . . . . . . . . . . 64
4.5.2 Transposition of Death Times . . . . . . . . . . . . . . . . 65
4.6 Characterization of Neuronal Morphologies . . . . . . . . . . 66

**Bibliography** **69**

# Chapter 1

# Introduction

Understanding the functionalities and structures of the mammalian brain has always been of great interest to scientists. At the core of this research has been the analysis of its fundamental units: neurons. Neurons are the cells responsible for receiving electric inputs from the environment and sending new electrical signals to target cells. They are composed of the *dendrites*, the *axon* and the *soma*, which contains the nucleus.

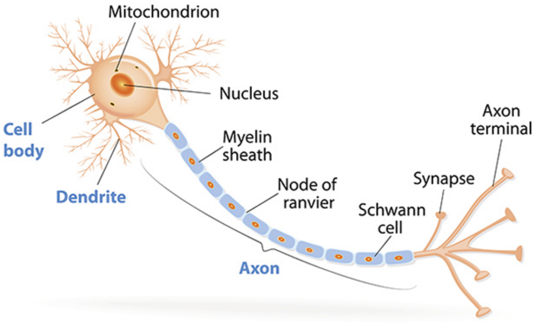


**Figure 1.1:** Illustration of a neuron with the soma, the axon and the dendrites [25].

Researchers have demonstrated that the shape of a neuron influences its functionality. Traditionally, neurons were observed under the microscope and neuroscientists classified them by their branching structure or total length.

It is possible to create a digital reconstruction of a neuron by sampling a set of points in $\mathbb{R}^3$ along each branch, together with edges connecting adjacent sets of points. This reconstruction is a combinatorial merge tree preserving the same morphological information as the neuronal cell. The dendrites correspond to the branches, the axon to the central branch and the soma to the root of the merge tree, and the height function is given by the Euclidean distance of each vertex from the root. Formally, a merge tree $(T, h)$ is a rooted combinatorial tree[1] equipped with a height function $h\colon V(T) \to \mathbb{R} \cup \{\infty\}$.

The development of an efficient technique to examine branching structures has proven more complex than expected. This issue has been addressed using tools from *Topological Data Analysis* (TDA). TDA is a field of data science whose goal is to capture the qualitative properties of the shape of finite metric spaces. The foundational tool of TDA is called *persistent homology*. Persistent homology is an invariant of finite metric spaces or more generally, filtered topological spaces, that encodes the information contained in the shape of data sets over multiple scales. It tracks the births and deaths of topological features across different scales. The result is a *barcode*, that provides valuable information about the structure of a given metric space or a filtered topological space.

The structure of a merge tree can be analyzed with the help of *sublevel set filtrations*, which are particularly useful for the computation of persistent homology. A sublevel set filtration is a nested sequence of spaces $\{X_{a_i}\}_{0 \leqslant i \leqslant n}$ where $a_0 \leqslant \ldots \leqslant a_n$ and $X_{a_i} = h^{-1}(-\infty, a_i]$ for $0 \leqslant i \leqslant n$. Intuitively, each sublevel set $X_{a_i} = h^{-1}[-\infty, a_i)$ represents a set of points with $h$-values less than $a_i$. To illustrate this, consider Figure 1.2, which depicts a merge tree $(T, h)$ along with a corresponding sublevel set filtration $\{h^{-1}(-\infty, a_i]\}_{1 \leqslant i \leqslant 3}$, where the values $a_i$ are taken from the set$\{2.5, 3.5, \infty\}$ for $1 \leqslant i \leqslant 3$. Each of these values defines a different sublevel set, corresponding to a distinct subset of the vertex set $V(T) = \{b_0, b_1, b_2, d_2, d_1, b_3, d_3, d_0\}$.

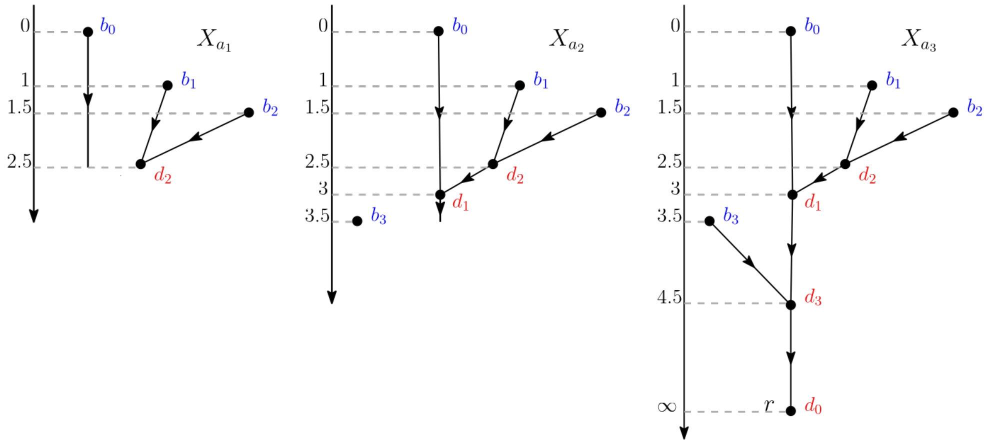


**Figure 1.2:** Example of a sublevel set filtration $\{X_{a_i}\}_{1 \leqslant i \leqslant 3}$ of a merge tree $(T, h)$, where $X_{a_3} = T$.

An example of the application of persistent homology to a merge tree $(T, h)$

---

[1] A rooted combinatorial tree is a tree with a distinguished vertex $r$, called the root. Each vertex in a combinatorial tree has either degree 3 or degree 1.

and the resulting barcode $B$ is depicted in Figure 1.3. The barcode $B$ is composed of four bars, each representing the lifetime of a different branch. For example the bar $[b_2, d_2)$ represents the branch with birth time $b_2$ and death time $d_2$. The longest bar $[b_0, d_0)$ is the trunk of the merge tree, to which the other branches are attached. Its birth time is $b_0$, which is the leaf with the lowest $h$-value and its death time is $d_0$, which is the vertex corresponding to the root. This visualization gives valuable information about the structure of the merge tree.

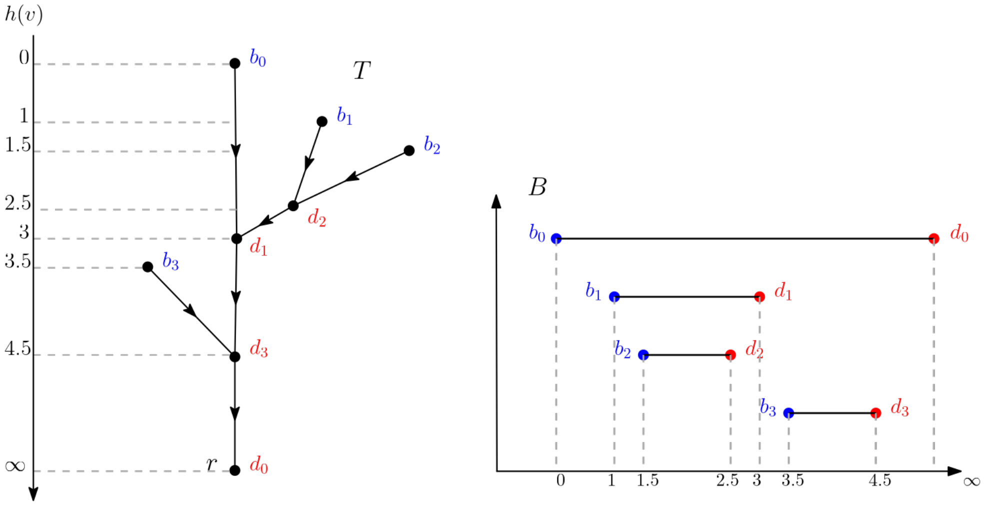


**Figure 1.3:** Application of persistent homology on a merge tree $(T, h)$ and the resulting barcode $B$.

Persistent homology is a key component for a stable and efficient algorithm for the automatic digital classification of neuronal morphologies. This algorithm is called *Topological Morphology Descriptor* (TMD) and is based on the application of persistent homology to merge trees to capture their shape. The TMD encodes the branching structure of neuronal trees in a persistent barcode as demonstrated in the previous example. Each interval of the persistent barcode represents the lifetime of a branch, containing its birth and death time. We compare the different tree structures with the use of the *bottleneck distance* [17, 15].

In *A Topological Representation of Branching Neuronal Morphologies* [16], it was demonstrated that the TMD algorithm can classify any type of a rooted tree equipped with a height function, providing a topological benchmark for the comparison of different structures. The TMD algorithm is applied with the goal of classifying pyramidal cells, which are a particular type of neuron associated with advanced cognitive functions found in the cerebral cortex of most mammalian brains. Using the TMD algorithm it was possible to classify pyramidal cells of different species by comparing the corresponding barcodes.

The resulting differences yielded a coherent classification, confirming the success rate of the TMD algorithm.(Figure 1.4)

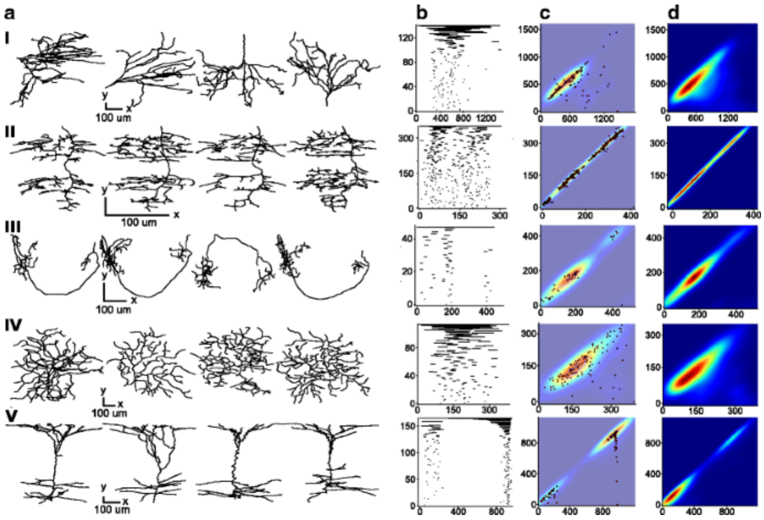


**Figure 1.4:** Comparison of pyramidal cells coming from differ species: Part (a) of the figure illustrates neurons from different species, each row corresponds to a species: (I) cat, (II) dragonfly, (III) fruit fly, (IV) mouse, (V) rat. In parts (b) and (c) there are respectively their corresponding persistent barcode and diagram and in (d) we can see an illustration of their unweighted persistent image [16].

The development of the TMD algorithm leads to the question of whether it is possible to recover the initial data, the neuron, from the barcode resulting from the TMD. This can be done by defining an algorithm called the *Topological Neuronal Synthesis* (TNS), which synthesizes an artificial neuron from the input barcode. The TNS consists of three components: *initiation of growth, elongation* and *branching/termination*. After the initiation of the growth, each growing tip is assigned a probability to bifurcate, terminate or elongate that depends on the distance from the soma. The morphology of the resulting artificial neurons is proved to be combinatorial equivalent to the digital reconstruction of the corresponding neurons, i.e. the digital reconstruction of the tree and the resulting artificial tree are isomorphic [17, 16].

In this thesis, we focus on the development of the TMD and TNS algorithms. To this end, we first introduce trees, simplicial complexes and homology. We then extend the concept of homology to finite metric spaces and filtered topological spaces by introducing persistent homology. We also analyze the behaviour and stability of the TNS algorithm, and show that the TNS beheaves as a stochastic right inverse for the TMD.

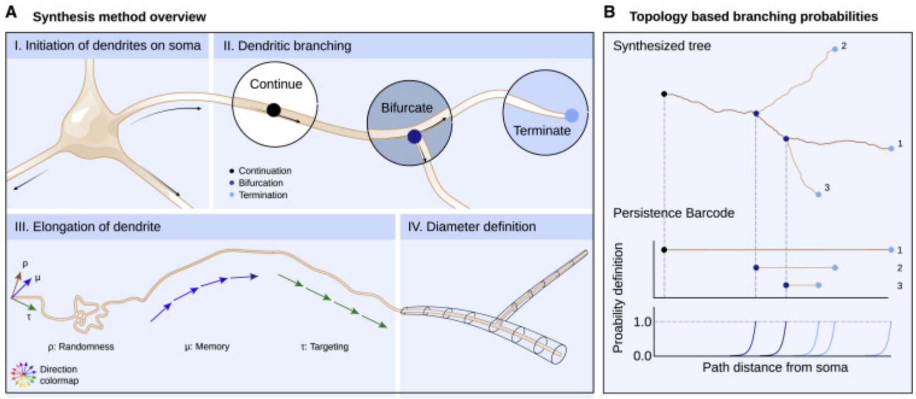


**Figure 1.5:** The different components of the TNS algorithm applied for the synthesis of a neuron in (A) and the growth of a neuronal tree by a given barcode in (B) with a graph of the associated probability to bifurcate or terminate [18].

## Chapter 2

# Trees, Simplicial Complexes and Homology

In this chapter, we review the definitions of graphs, trees and simplicial complexes, and define homology groups. Before defining trees with the help of [6, 17], we introduce the basic notions of graphs using the book *Graph Theory* [8] written by Diestel, Reinhard. For the introduction of simplicial complexes and homology theory, we refer to *Algebraic Topology* [14], *Computational topology for data analysis* [7] and *Topological pattern recognition for point cloud data* [3].

## 2.1 Graphs

The first objects we introduce are graphs, which are mathematical objects used to model various types of relations.

**Definition 2.1** *A **graph** is a pair* $G = (V(G), E(G))$ *of sets, such that*

$$E(G) \subseteq V(G) \times V(G).$$

*We always assume that* $V(G) \cap E(G) = \emptyset$*. The elements of* $V(G)$ *are called **vertices** of the graph* $G$ *and the elements of* $E(G)$ *are its **edges**. The graph* $G$ *is said to be **finite** if its number of vertices is finite.*

**Example 2.2** *In Figure 2.1 there are two graphs* $G$ *and* $G'$*, where the respective edges and vertices are given by*

$$\begin{aligned} V(G) &= \{v_0, v_1, v_2, v_3\}, \qquad & E(G) &= \{\{v_0, v_1\}, \{v_1, v_2\}, \{v_1, v_3\}, \{v_2, v_3\}\}, \\ V(G') &= \{w_0, w_1, w_2, w_3\}, \qquad & E(G') &= \{\{w_0, w_1\}, \{w_1, w_2\}, \{w_2, w_3\}\}. \end{aligned}$$

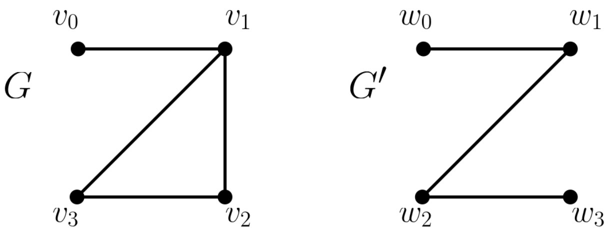


**Figure 2.1:** The picture on the left depicts a graph $G$ with vertex set $V(G) = \{v_0, v_1, v_2, v_3\}$ and edges $E(G) = \{\{v_0, v_1\}, \{v_1, v_2\}, \{v_1, v_3\}, \{v_2, v_3\}\}$. The one on the right right depicts a graph $G'$ with vertex set $V(G') = \{w_0, w_1, w_2, w_3\}$ and edges $E(G') = \{\{w_0, w_1\}, \{w_1, w_2\}, \{w_2, w_3\}\}$.

**Definition 2.3** *For a graph $G$ two vertices $v, w \in V(G)$ are called* ***adjacent*** *vertices if $\{v, w\} \in E(G)$ is an edge of $G$. The* ***degree*** *of a vertex $v \in V$ is the number of edges containing $v$, i.e.*

$$\deg(v) := |\{w \in V(G) \mid \{v, w\} \in E(G)\}|.$$

**Definition 2.4** *A graph $G$ is called a* ***binary graph*** *if for every vertex $v \in V(G)$, $deg(v)$ is either $1$ or $3$.*

**Definition 2.5** *A* ***labeling of a graph*** *$G$ is a map $\rho : V(G) \to S$, where $S$ is a set of labels. If $S$ is a subset of the natural numbers $\mathbb{N}$, then we call the map $\rho$ an ordered labeling.*

**Definition 2.6** *A* ***path*** *is a non-empty graph $P = (V(P), E(P))$ of the form*

$$V(P) = \{v_0, \ldots, v_k\}, \quad E(P) = \{v_0v_1, \ldots, v_{k-1}v_k\},$$

*where the $v_i$ are all distinct. If $P = v_0 \ldots v_{k-1}$ is a path and $k \geqslant 3$, then the graph $C := P + v_{k-1}v_0$ is called a* ***cycle****. We call a graph with no cycles an* ***acyclic graph****. The number of edges of a path or of a cycle is its* ***length****.*

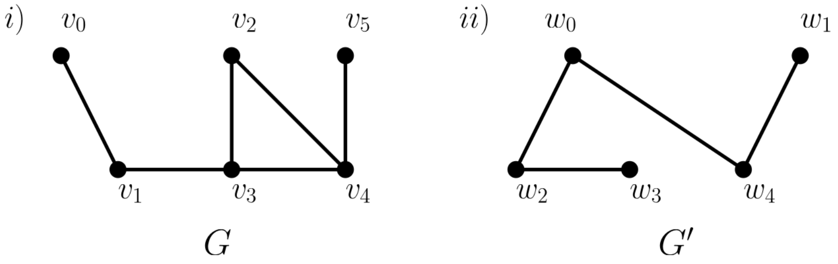


**Figure 2.2:** The graph in $(i)$ has one cycle, denoted by $v_3v_2v_4v_3$ with length 3 and the graph in $(ii)$ is an example of an acyclic graph.

**Remark 2.7** *Intuitively a path is a sequence of adjacent vertices $v_0, \ldots, v_k$, and a cycle is a path with $v_i \neq v_j, i \neq j$, for $1 \leqslant i, j \leqslant k-1$, for which $v_1 = v_k$.*

**Definition 2.8** *The* ***distance*** $d_G(v,w)$ *in* $G$ *between two vertices* $v,w \in V(G)$ *is the length of the shortest path between* $v$ *and* $w$ *in* $G$*. If such a path does not exist, we set* $d_G(v,w) := \infty$*.*

To give a purely combinatorial representation of a finite graph, we introduce **adjacency matrices**. These are useful to verify whether two vertices are adjacent or not.

**Definition 2.9** *The* ***adjacency matrix*** $A = (a_{ij})_{n\times n}$ *of* $G$ *is defined by*

$$a_{ij} := \begin{cases} 1 & \text{if } v_iv_j \in E(G), \\ 0 & \text{otherwise.} \end{cases}$$

**Remark 2.10** *Adjacency matrices exist for all graphs that have an ordered labeling of the vertices.*

**Example 2.11** *For the graph in Figure 2.2* $(i)$*, the adjacency matrix is*

$$A(G) = \begin{array}{c} \\ v_0 \\ v_1 \\ v_2 \\ v_3 \\ v_4 \\ v_5 \end{array} \begin{array}{c} \begin{array}{cccccc} v_0 & v_1 & v_2 & v_3 & v_4 & v_5 \end{array} \\ \begin{bmatrix} 0 & 1 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 1 & 0 \\ 0 & 1 & 1 & 0 & 1 & 0 \\ 0 & 0 & 1 & 1 & 0 & 1 \\ 0 & 0 & 0 & 0 & 1 & 0 \end{bmatrix} \end{array}.$$

**Definition 2.12** *Let* $G = (V(G), E(G))$ *and* $G' = (V(G'), E(G'))$ *be two graphs, then a map* $\varphi : V(G) \to V(G')$ *is a* ***graph homomorphism*** *from* $G$ *to* $G'$ *if it preserves the adjacency of the vertices, i.e.* $\{x,y\} \in E(G)$ *implies that* $\{\varphi(x), \varphi(y)\} \in E(G')$*. If* $\varphi$ *is bijective and its inverse* $\varphi^{-1}$ *is also a homomorphism, we call* $\varphi$ *a* ***graph isomorphism*** *and say that* $G$ *and* $G'$ *are isomorphic,* $G \sim G'$*.*

**Example 2.13** *In Figure 2.3 the map* $\varphi\colon V(G) \to V(G')$ *given by*

$$\begin{aligned} \varphi(v_0) &= w_1, & \varphi(v_1) &= w_1, & \varphi(v_2) &= w_4, \\ \varphi(v_3) &= w_3, & \varphi(v_4) &= w_2, & \varphi(v_5) &= w_5. \end{aligned}$$

*defines a graph isomorphism.*

**Remark 2.14** *If two graphs are isomorphic, then their adjacency matrices are equal. Thus, we do not make any distinction between isomorphic graphs and write* $G = G'$ *instead of* $G \sim G'$*. This distinction is only considered in case one refers to* $G$ *and* $G'$ *as abstract graphs.*

**Definition 2.15** *A graph* $G$ *is called* ***connected*** *if it is non-empty and any two of its vertices are linked by a path in* $G$*.*

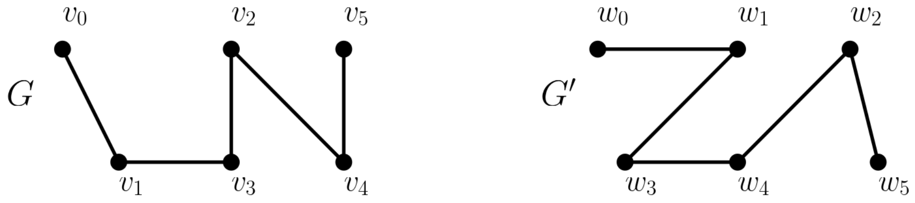
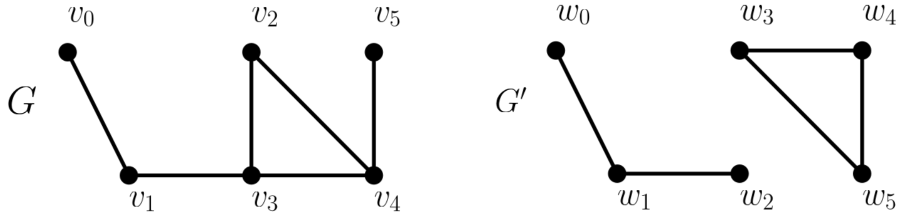


**Figure 2.3:** The graphs $G$ and $G'$ are isomorphic through $\varphi$.



*i)* A connected graph. *ii)* A disconnected graph.

**Figure 2.4:** A connected and a disconnected graph.

**Definition 2.16** *A **directed graph** is a pair G of vertices and edges together with two maps*

$$init\colon E(G) \to V(G), \quad ter\colon E(G) \to V(G).$$

*The maps init and ter assign to every edge e an initial vertex init(e) and terminal vertex ter(e). The edge e is said to be directed from init(e) to ter(e).*

**Example 2.17** *In Figure 2.5, G is a directed graph with vertices and edges*

$$V(G) = \{v_0, v_1, v_2, v_3, v_4\} \text{ and } E(G) = \{e_0, e_1, e_2, e_3, e_4, e_5\}.$$

*Notice that every edge $e_i \in E(G)$ has a corresponding $init(e_i)$ and $ter(e_i)$.*

$$\begin{aligned} &e_0 : init(e_0) = v_0 \text{ and } ter(e_0) = v_1, && e_1 : init(e_1) = v_0 \text{ and } ter(e_1) = v_2, \\ &e_2 : init(e_2) = v_2 \text{ and } ter(e_2) = v_3, && e_3 : init(e_3) = v_1 \text{ and } ter(e_3) = v_3, \\ &e_4 : init(e_4) = v_4 \text{ and } ter(e_4) = v_3, && e_5 : init(e_5) = v_3 \text{ and } ter(e_5) = v_4. \end{aligned}$$

**Definition 2.18** *For a directed graph G, we call a vertex $w \in V(G)$ **adjacent to** $v \in V$, if $(v, w) \in E(G)$ or if $(w, v) \in E(G)$. The elements of*

$$N^+(v) := \{w \in V(G) \mid (v, w) \in E(G)\}, \quad N^-(v) := \{u \in V \mid (u, v) \in E(G)\}$$

*are respectively called the **successors** of v and the **predecessors** of v. The **in-degree** is defined as $deg^+(v) := |N^+(v)|$ and the **out-degree** as $deg^-(v) := |N^-(v)|$. The **degree** of a vertex v is given by*

$$deg(v) := deg^+(v) + deg^-(v).$$

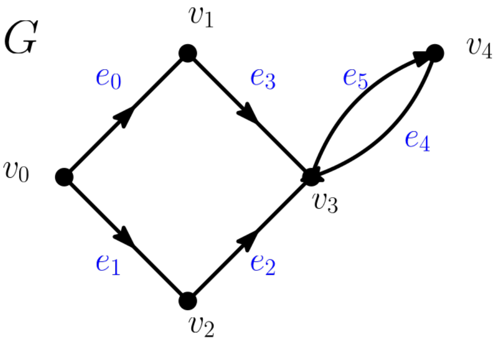


**Figure 2.5:** Example of a directed graph $G$ with vertices $V(G) = \{v_0, v_1, v_2, v_3, v_4\}$ and edges $E(G) = \{(v_0, v_1), (v_0, v_2), (v_1, v_3), (v_2, v_3), (v_1, v_3), (v_3, v_4), (v_4, v_3)\} = \{e_0, e_1, e_2, e_3, e_4, e_5\}$.

**Example 2.19** *Consider the directed graph $G$ in Figure 2.5. Successors and predecessors of the vertex $v_3$ are*

$$N^+(v_3) = \{w \in V(G) \mid (v_3, w) \in E\} = \{v_4\},$$
$$N^-(v_3) = \{w \in V(G) \mid (w, v_3) \in E\} = \{v_1, v_2, v_4\}.$$

*Now that we have successors and predecessors, we can calculate the degree of $v_3$. Notice that a vertex can be both in $N^+(v_3)$ and $N^-(v_3)$.*

$$deg^+(v_3) = |N^+(v_3)| = 1, \qquad deg^-(v_3) = |N^-(v_3)| = 3,$$
$$deg(v_3) = deg^+(v_3) + deg^-(v_3) = 4.$$

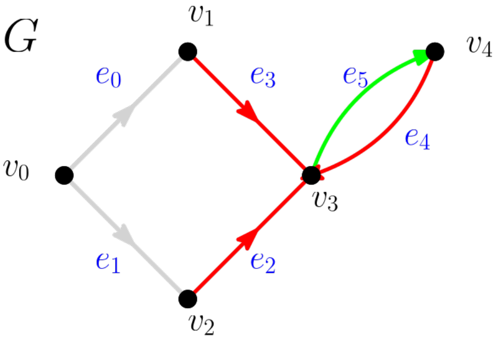


**Figure 2.6:** The directed graph in Figure 2.5 with edges between the predecessors of the vertex $v_3$ in red and successors in green.

## 2.2 Trees

To create a digital model of a neuron we rely on the notion of rooted trees. In particular, we focus on binary trees, since the probability that two branches of a neuron bifurcate at the exact same point is almost zero. In this section we use the notions and definitions introduced in the papers *From trees to barcodes and back again I, II* [17, 6].

### 2.2.1 Combinatorial Trees

**Definition 2.20** *A* ***combinatiorial tree*** *T is a connected, acyclic, binary, directed graph, such that each vertex has either degree* 3*, called an* ***inner vertex****, or degree* 1*, called a* ***leaf****. A combinatorial tree T is* ***finite*** *if the number of vertices is finite. A vertex v is a* ***parent*** *of a vertex w if there exists a directed edge from w to v, in that case, w is a* ***child*** *of v.*

**Remark 2.21** *In a combinatorial tree, there are no vertices with degree* 2*.*

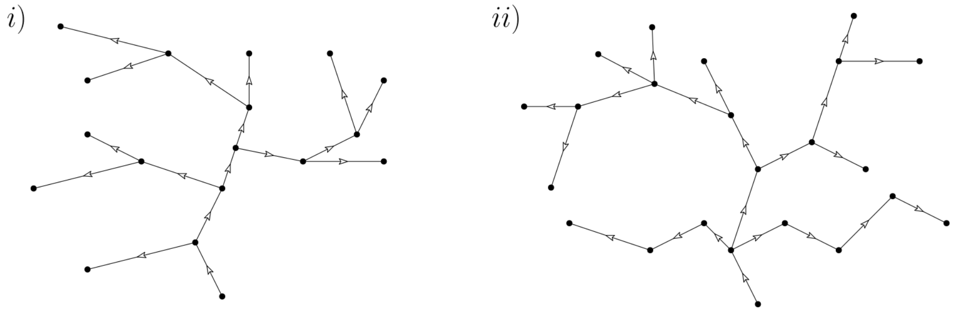


**Figure 2.7:** A combinatorial tree in $(i)$ and a non-combinatorial tree in $(ii)$.

**Example 2.22** *In Figure 2.7 the tree in* $(i)$ *is a combinatorial tree since each vertex has either degree* 1 *or degree* 3*. On the other hand, the object in* $(ii)$ *is not a combinatorial tree since there are vertices that have degree* 2*.*

**Definition 2.23** *A* ***rooted tree*** *T is a combinatorial tree with a distinguished vertex r called the* ***root****. This is the only vertex of degree* 1 *in T that has no parent.*

**Remark 2.24** *In the case of a combinatorial tree an inner vertex is also called a* ***branching*** *or a* ***bifurcation point*** *and a leaf is called a* ***termination****. Every vertex, besides the root, has a unique parent and at most two children.*

It is also important to note that every tree is fully defined by the set of vertices $V$ and the partial order "is a parent of".

**Remark 2.25** *Since combinatorial trees are a special case of graphs all the definitions for graphs introduced in Section 2.1 still hold for combinatorial trees.*

**Proposition 2.26** *For a rooted combinatorial tree T an embedding of T in* $\mathbb{R}^3$ *always exists.*

**Proof** Every rooted combinatorial tree with root $r$ is defined by its set of vertices $V(T)$ and its parent-child relations. Regarding the set of vertices

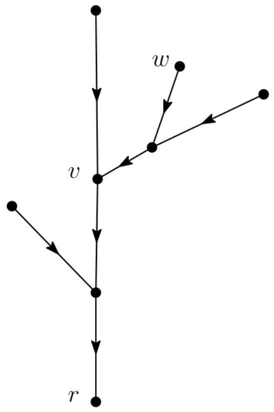


**Figure 2.8:** A combinatorial rooted tree with root $r$. The vertices $v$ and $w$ are respectively an example of a *branching point* and of a *termination*. Moreover, the directed edge from $w$ to $v$ tells us that $v$ is a parent of $w$.

as a subset of $\mathbb{R}^3$ enables to place the vertices following the parent-child relation, where parents are placed above children. Start from the root, which is always placed at $(0,0,0)$ and continue by assigning coordinates $(x,y,z)$ to each vertex $v \in V(T)$, where the $z$-coordinate depends on the tree depth of $v$ and the $x$- and $y$- coordinate are chosen such that no vertices overlap. $\square$

In this thesis, we mostly focus on **geometric trees**, which are the type of trees used for modeling neurons. We also refer to them as **neuronal trees**.

**Definition 2.27** *A* ***geometric tree*** *is the embedding of a combinatorial tree in* $\mathbb{R}^3$. *The set of geometric trees is denoted by* $\mathcal{T}$.

On the set of geometric trees we can define the following equivalence relation. Let $T, T' \in \mathcal{T}$,

$$T \underset{comb}{\sim} T' \iff T \text{ and } T' \text{ are embeddings of the same finite rooted tree.}$$

### 2.2.2 Merge Trees

Trees were introduced to study the relations between people (*family trees*) and/or the relations between species (*phylogenetic trees*). This is the reason for the common use of the words *parent* and *child*. The relation "is a parent of" is mostly described in mathematics by a **height function**, which gives rise to the notion of **merge trees**.

**Definition 2.28** *A* ***merge tree*** *is a rooted combinatorial tree $T$ together with a function $h\colon V(T) \to \mathbb{R} \cup \{\infty\}$, called a* ***height function*** *that satisfies two properties.*

1. *If $v$ is the parent of $w$, then $h(v) \geqslant h(w)$,*
2. *if $r$ is the root node, then $h(r) = \infty$.*

*Two merge trees $(T, h)$ and $(T', h')$ are* ***isomorphic*** *if there is a graph isomorphism $\varphi\colon T \to T'$ such that $h = h' \circ \varphi$. A* ***generic merge tree*** *is a merge tree $(T, h)$, such that $h$ is injective.*

**Example 2.29** *The tree $T$ in Figure 2.9 is an example of a merge tree, with height function $h\colon V(T) \to \mathbb{R} \cup \{\infty\}$. The vertices of $T$ are*

$$V(T) = \{b_0, b_1, b_2, b_3, d_0, d_1, d_2, d_3\}.$$

*Take now $b_3, d_3 \in V(T)$, then $h(d_3) \leqslant h(b_3)$ since $d_3$ is a parent of $b_3$. It also holds that $h(r) = \infty$, where $r$ is the root of $T$.*

**Remark 2.30** *Trees are normally drawn with the root higher than the leaves. We do not use this convention and consider trees with a similar structure as trees that appear in nature, where the root is lower than the leaves.*

**Proposition 2.31** *Every merge tree $(T, h)$ has an ordered labeling.*

**Proof** Consider the height function $h\colon V \to \mathbb{R} \cup \{\infty\}$ and order the vertices according to their $h$-value. Assign to the leaf $l$ with the lowest $h$-value the label 0. Then label the remaining vertices based on their order of appearance.□

**Definition 2.32** *Labels on the leaves are called* ***birth labels*** *and the ones on the internal vertices are called* ***death labels****.*

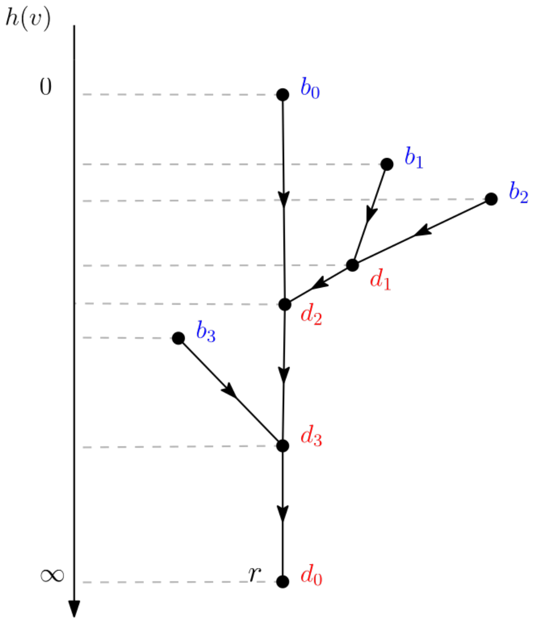


**Figure 2.9:** A merge tree, where the vertices are labeled according to their order of appearance. Moreover, the labels in blue are the birth labels and the ones in red are the death labels.

**Definition 2.33** *Two generic merge trees* $(T, h)$ *and* $(T', h')$ *are* ***combinatorially equivalent*** *if there exists a graph isomorphism* $\varphi\colon T \to T'$*, such that:*

1. *For every pair of leaf nodes* $v_i, v_j \in V(T)$*, if* $h(v_i) < h(v_j)$*, then*
$$h'(\varphi(v_i)) < h'(\varphi(v_j)).$$
2. *For every pair of internal nodes* $v_i, v_j \in V(T)$*, if* $h(v_i) < h(v_j)$*, then*
$$h'(\varphi(v_i)) < h'(\varphi(v_j)).$$

**Definition 2.34** *A* ***combinatorial merge tree*** $T$ *is a combinatorial tree equipped with an order labeling* $L_l$ *of the leaves and* $L_i$ *of the internal nodes such that for internal nodes* $v, w \in V(T)$*, if* $v$ *is a parent of* $w$*, then* $L_i(v) > L_i(w)$*.*

## 2.3 Simplicial Complexes

In this section, we introduce simplicial complexes, which generalize graphs. They are combinatorial objects built by gluing vertices, edges, faces, etc. along common boundaries. With the help of these combinatorial objects,we approximate topological spaces, including finite metric spaces.

**Definition 2.35** *For a finite set of* $k+1$ *points* $X = \{x_0, \ldots, x_k\} \in \mathbb{R}^d$ *we say that* $x = \sum_{i=0}^{k} t_i x_i$ *is an* ***affine combination****, if* $\sum_{i=0}^{k} t_i = 1$*. The set of all affine combinations is called* ***affine hull****. It is a* $k$***-plane*** *if the* $k+1$ *points are* ***affinely independent****, by which we mean that any two affine combinations* $x = \sum_{i=0}^{k} t_i x_i$ *and* $y = \sum_{i=0}^{k} s_i x_i$ *are equal if and only if* $t_i = s_i$ *for every* $1 \leqslant i \leqslant k$*.*

**Remark 2.36** *For a d-dimensional space* $\mathbb{R}^d$ *we can have at most d linearly independent vectors and hence* $d+1$ *affinely independent points.*

**Definition 2.37** *An affine combination* $x = \sum_{i=0}^{k} t_i x_i$ *is a* ***convex combination*** *if* $t_i \geqslant 0$ *for each* $1 \leqslant i \leqslant k$*. Moreover, the set of all convex combinations is called a* ***convex hull****.*

**Definition 2.38** *Let* $X = \{x_0, \ldots, x_k\}$ *be a set of affinely independent* $k+1$ *points. A* $k$***-simplex*** *is the convex hull of* $k+1$ *affinely independent points* $\sigma = \sigma\{x_0, \ldots, x_k\}$*. This is also called the* ***simplex spanned by*** $X$ *in* $\mathbb{R}^k$ *and its dimension is* $\dim \sigma = k$*. The points* $x_i$ *are called* ***vertices*** *and for any subset* $\emptyset \neq Y \subseteq X$ *we call the simplices* $\sigma(Y)$*, spanned by* $Y$*, the* ***faces*** *of* $\sigma$*. Since* $Y$ *is a subset of an affinely independent set* $X$*, it is also affinely independent and hence defines a simplex.*

**Example 2.39** *Embeddings of graphs, introduced in Section 2.1, are a special case of simplicial complexes.*

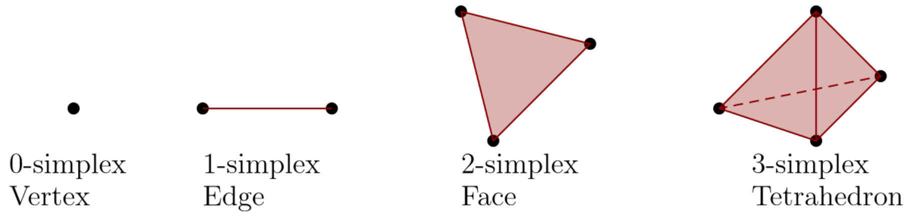


**Figure 2.10:** A graphic representation of a vertex, an edge a face, and a tetrahedron.

**Remark 2.40** *The simplices of the lowest dimension have special names. For example, a* 0*-simplex is called a* ***vertex****, a* 1*-simplex an* ***edge****, a* 2*-simplex a* ***triangle*** *and a* 3*-simplex a* ***tetrahedron****.*

Gluing together simplices of different dimensions along common boundaries leads to the notion of geometric and abstract simplicial complexes.

**Definition 2.41 (Geometric Simplicial Complex)** *A* ***geometric simplicial complex*** *is a finite collection $\mathcal{X}$ of simplices in a Euclidean space such that the following conditions hold:*

1. *For any simplex $\sigma \in \mathcal{X}$, all faces of $\sigma$ are also contained in $\mathcal{X}$,*
2. *For any two simplices $\sigma$ and $\tau$ of $\mathcal{X}$, the intersection $\sigma \cap \tau$ is a simplex, which is a face of both $\sigma$ and $\tau$.*

*The dimension of $\mathcal{X}$ is $\dim \mathcal{X} = \max\{\dim \sigma \mid \sigma \in \mathcal{X}\}$.*

**Example 2.42** *Figure 2.11 $(i)$, $(ii)$ represents two examples of geometric simplicial complexes. In contrast, in Figure $(iii)$ and $(iv)$ we have two objects that are not geometric simplicial complexes. In $(iii)$ we can see that the yellow triangle is not a face of any of the two triangles in contradiction to $(2)$ of Definition 2.41. In $(iv)$ we have a tetrahedron with a missing face, which does not fulfill requirement $(1)$ of the definition.*

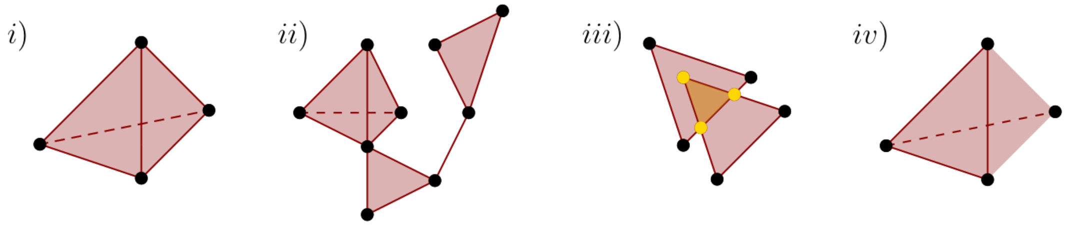


**Figure 2.11:** Object in $(i)$ and $(ii)$ are examples of simplicial complex, but $(iii)$ and $(iv)$ are not.

Forgetting about the geometry of simplicial complexes and focussing only on the connection between their vertices provides a combinatorial construction, known as *abstract simplicial complexes*. These complexes are a generalization of graphs that include higher dimensional simplices, like triangles and tetrahedra. Abstract simplicial complexes are easier to construct than geometric simplicial complexes because it is not necessary to provide an embedding in an Euclidean space.

**Definition 2.43 (Abstract Simplicial Complex)** *An* ***abstract simplicial complex*** *$X$ is a pair $X = (V(X), \Sigma(X))$, where $V(X)$ is a finite set called the* ***vertices*** *of $X$ and $\emptyset \neq \Sigma(X)$, called the* ***simplices*** *is a subset of the collection of all subsets of $V(X)$ satisfying the condition:*

$$\textit{If } \sigma \in \Sigma(X) \textit{ and } \emptyset \neq \tau \subseteq \sigma, \textit{ then } \tau \in \Sigma(X).$$

*We say that a $k$-simplex $\sigma$ has dimension $k$ if $|\sigma| = k + 1$.*

**Definition 2.44** *Let $X_1, X_2$ be abstract simplicial complexes with vertex sets $V_1, V_2$, respectively. An* ***isomorphism*** *between $X_1, X_2$ is a bijection $\varphi\colon V_1 \to V_2$ such that the sets $X_1$ and $X_2$ are the same under the renaming of the vertices by $\varphi$ and its inverse.*

Comparing the definition of geometric and abstract simplicial complexes, the existence of a correspondence between the two is almost evident. Note that given a geometric simplicial complex, it is always possible to construct an abstract simplicial complex by ignoring all simplices of dimension bigger than 0 and keeping only its set of vertices. In this way, any simplicial complex $\mathcal{X}$ determines an abstract simplicial complex $X$, called a **vertex scheme** of $\mathcal{X}$.

**Definition 2.45 (Vertex Scheme)** *Let $\mathcal{X}$ be a geometric simplicial complex with vertices $V$ and let $X$ be the collection of all subsets $\{v_0, \dots, v_k\}$ of $V$ such that the vertices $\{v_0, \dots, v_k\}$ span a simplex of $\mathcal{X}$. The collection $X$ is called the* ***vertex scheme*** *of $\mathcal{X}$.*

We additionally notice that every abstract simplicial complex $X$ determines the underlying space of the geometric simplicial complex in $\mathbb{R}^d$, for $d$ sufficiently large, up to homeomorphism. This is denoted by $|X|$ and is called the **geometric realization** of $X$.

**Definition 2.46** *A geometric simplicial complex $|X| \subseteq \mathbb{R}^d$ is called a* ***geometric realization*** *of an abstract simplicial complex $X$ if and only if there exists an embedding $e\colon V(X) \to \mathbb{R}^m$ that takes every $k$-simplex $\{x_0, \dots, x_k\}$ in $X$ to a $k$-simplex in $|X|$ that is the convex hull of $\{e(x_0), \dots, e(x_k)\}$. It is uniquely determined up to isomorphism.*

The geometric realization of an abstract simplicial complex $K$ with $n$ vertices as an $(n-1)$-simplex in $\mathbb{R}^{n-1}$ can always be constructed. To do this, consider an $(n-1)$-simplex in $\mathbb{R}^n$, where each vertex $v_i$ has the $i$-th coordinate equal to 1 and all the other equal to 0.

**Example 2.47** *Observe the pair* $X = (V(X), \Sigma(X))$, *with*

$$\begin{aligned} V(X) &= \{v_0, v_1, v_2, v_3, v_4\}, \\ \Sigma(X) &= \{V(X), [v_0, v_1], [v_0, v_2], [v_1, v_2], [v_0, v_3], [v_1, v_3], [v_2, v_3], [v_2, v_4], \\ &\quad [v_3, v_4], [v_0, v_1, v_3], [v_1, v_2, v_3], [v_0, v_1, v_2], [v_0, v_2, v_3], [v_0, v_2, v_1, v_3]\}. \end{aligned}$$

*This is an abstract simplicial complex, since* $V(X)$ *is finite,* $\Sigma(X)$ *is not empty and it holds that for all* $\sigma \in \Sigma(X)$ *with* $\emptyset \neq \tau \subseteq \sigma, \tau \in \Sigma(X)$. *The geometric realization of the abstract simplicial complex* $X$ *is represented in Figure 2.12. Now consider* $X' = (V(X'), \Sigma(X'))$ *with*

$$\begin{aligned} V(X') &= \{v_0, v_1, v_2, v_3, v_4\}, \\ \Sigma(X') &= \{V(X'), [v_0, v_1], [v_0, v_2], [v_1, v_2], [v_0, v_3], [v_1, v_3], [v_2, v_3], [v_2, v_4], \\ &\quad [v_3, v_4], [v_0, v_1, v_3], [v_0, v_1, v_2], [v_0, v_2, v_3], [v_0, v_2, v_1, v_3]\}. \end{aligned}$$

*Then* $[v_1, v_2, v_3] \notin \Sigma(X')$, *but* $[v_1, v_2, v_3] \subseteq [v_1, v_2, v_3, v_4]$, *hence* $X'$ *can not be an abstract simplicial complex.*

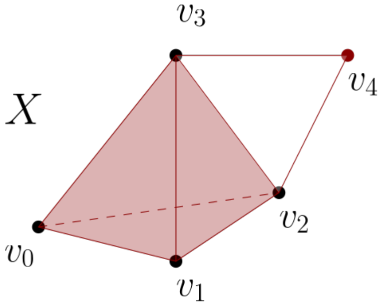


**Figure 2.12:** Geometric realization of $X$ in Example 2.47.

**Definition 2.48** *Let* $X$ *and* $Y$ *be two abstract simplicial complexes. A* ***map of abstract simplicial complexes*** $f\colon X \to Y$ *is a map of sets* $f_V\colon V(X) \to V(Y)$ *such that for any simplex* $\sigma \in \Sigma(X)$, *it holds* $f_V(\sigma) \in \Sigma_Y$.

The geometric realization construction is functorial, which means that every map $f\colon X \to Y$ induces a continuous map $|f|\colon |X| \to |Y|$ such that:

- $|f \circ g| = |f| \circ |g|$,
- $|id_X| = id_{|X|}$.

We can approximate a topological space by using **triangulation**, which means finding a geometric simplicial complex $K$ homeomorphic to $X$. Triangulable spaces include differentiable manifolds and topological manifolds of dimension less than three.

**Definition 2.49** *Let* $X$ *be a topological space. An abstract simplicial complex* $K$ *is called a* ***triangulation of*** $X$ *if there exists an homeomeorphism* $f\colon |K| \to X$, *where* $|K|$ *is the geometric realization of* $K$.

**Example 2.50** *Let $T^2 \subseteq \mathbb{R}^3$ be the torus. The Torus is triangulable, since it is a two-dimensional topological manifold. One possinle triangulation of the torus is depicted in Figure 2.13.*

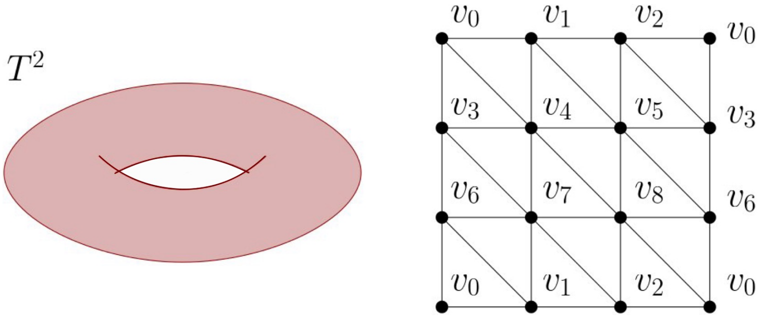


**Figure 2.13:** The two-dimensional torus (*left*) and one of its triangulations(*right*).

## 2.4 Homology

Homology groups are one of the most important tools of algebraic topology. They capture information about connectivity, voids, and holes within a topological space by analyzing boundaries and cycles. The idea behind homology is to count the number of cavities, by identifying all the cycles that are not a boundary of some subspace of $X$.

**Definition 2.51** *For a field $k$, let $X$ be a simplicial complex and $i \geqslant 0$. An **i-chain** is a formal sum of $i$-simplices in $X$ written as $c = \sum a_n \sigma_n$, where the $\sigma_n$ are the $i$-simplices and $a_n \in k$ are the coefficients.*

**Definition 2.52** *Let $X$ be a simplicial complex. The $i$-chains form a vector space over $k$ with the binary operations:*

- *Componetwise addition: Let $c = \sum a_n \sigma_n$ and $d = \sum b_n \sigma_n$ be $i$-chains, then*

$$c + d := \sum (a_n + b_n)\sigma_n.$$

- *Scalar multiplication: Let $\alpha \in k$ and $c = \sum a_n \sigma_n$ be an $i$-chain, then*

$$\alpha \cdot c = \alpha \cdot a \sum a_n \sigma_n := \sum \alpha a_n \sigma_n.$$

*The identity is the chain $0 = \sum 0 \cdot \sigma_n$ and the inverse of a chain $c$ is $-c$. This group is called the **group of i-chains** and is denoted $C_i = C_i(X)$.*

**Remark 2.53** *For dimensions $i$ less than zero and greater than $\dim X$ the group $C_i(X)$ is trivial.*

From now on we fix $X$ a simplicial complex and use the notation $C_i$ without specifying the complex.

To establish a connection between groups of different dimensions, we define maps $\partial_i\colon C_i \to C_{i-1}$ for every $i$.

**Definition 2.54** *Let $\sigma = [v_0, \dots, v_i] \in C_i$ be an i-simplex spanned by the vertices $\{v_0, \dots, v_i\}$. The* ***boundary*** *of $\sigma$ is given by*

$$\partial_i(\sigma) = \sum_{n=0}^{i} (-1)^n \sigma|_{[v_0,\dots,\hat{v_n},\dots,v_i]}.$$

*The boundary of the i-simplex $[v_0, \dots, v_i]$ is a sum of its $(i-1)$-dimensional faces, where $\hat{v}_j$, means that the vertex $v_j$ is omitted.*

**Remark 2.55** *If k is $\mathbb{Z}_2$, it is not important to take into account the factor $(-1)^n$ in the boundary of a simplex, since $1 = -1$.*

**Remark 2.56** *Note that for an i-chain $c = \sum a_n \sigma_n$, the boundary $\partial_i(c) = \sum a_n \partial_i(c)$ is a linear combination of $(i-1)$-chains.*

The map $\partial_i\colon C_i \to C_{i-1}$ is a homomorphism since

- $\partial_i(c + d) = \partial_i(c) + \partial_i(d)$ for every $i$-chain $c, d \in C_i$,
- $\partial_i(\alpha c) = \alpha \partial_i(c)$ for every $i$-chain $c \in C_i$ and for every $\alpha \in k$.

It is called the **boundary homomorphism** or boundary map.

**Proposition 2.57** *For the groups $C_i$ and the maps $\partial_i$, the identity*

$$\partial_{i-1} \circ \partial_i \equiv 0$$

*holds for every $i \geqslant 0$.*

**Proof** Let $\sigma \in C_i(X)$, then $\partial_i(\sigma) = \sum_{n=0}^{i} (-1)^n \sigma|_{[v_0,\dots,\hat{v_n},\dots,v_i]}$, we calculate

$$\begin{aligned}
\partial_{i-1}\partial_i(\sigma) &= \partial_{i-1}\Big(\sum_{n=0}^{i} (-1)^n \sigma|_{[v_0,\dots,\hat{v_n},\dots,v_i]}\Big) = \\
&= \sum_{m<n} (-1)^n (-1)^m \sigma|_{[v_0,\dots,\hat{v_m},\dots,\hat{v_n},\dots,v_i]} + \\
&+ \sum_{m>n} (-1)^n (-1)^{m-1} \sigma|_{[v_0,\dots,\hat{v_n},\dots,\hat{v_m},\dots,v_i]}
\end{aligned}$$

After switching $m$ and $n$ in the second sum, it becomes the negative of the first, hence the latter two summations cancel. ☐

**Example 2.58** *Consider the simplicial complex $X$ of Figure 2.14. First, we determine the chain groups that appear in $X$.*

$$\begin{aligned}
C_0(X) &= \{v_0, v_1, v_2, v_3, v_4\},\\
C_1(X) &= \{[v_0, v_1], [v_0, v_2], [v_1, v_2], [v_2, v_3], [v_2, v_4], [v_3, v_4]\},\\
C_2(X) &= \{[v_2, v_3, v_4]\}.
\end{aligned}$$

*For $i > 2$ we do not have simplices, hence $C_i(X) = 0$. Since the map $\partial_i$ is an homomorphism for every $i$ it holds that $\partial_i(0) = 0$. At this point, we are ready to compute the boundary of each element of the chain groups. For $C_0(X)$ it holds that $\partial_0(v_i) = 0$ for every $i \in \{0, \dots, 4\}$. For the edges, we have the following boundaries:*

$$\begin{aligned}
\partial_1([v_0, v_1]) &= v_1 - v_0, & \partial_1([v_0, v_2]) &= v_2 - v_0,\\
\partial_1([v_1, v_2]) &= v_2 - v_1, & \partial_1([v_2, v_3]) &= v_3 - v_2,\\
\partial_1([v_2, v_4]) &= v_4 - v_2, & \partial_1([v_3, v_4]) &= v_4 - v_3.
\end{aligned}$$

*The boundary of the triangle $[v_2, v_3, v_4]$ is given by*

$$\partial_2([v_2, v_3, v_4]) = [v_3, v_4] - [v_2, v_4] + [v_2, v_3].$$

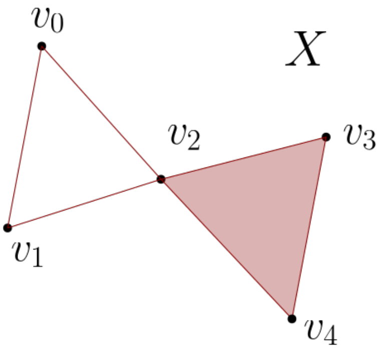


**Figure 2.14:** Simplicial complex of Example 2.58

**Definition 2.59** *A **chain complex** $C_\bullet$ over a field $k$ is a sequence of abelian groups $C_i$ for $i \geqslant 0$ together with linear maps $\partial_i \colon C_i \to C_{i-1}$ , such that $\partial_{i-1} \circ \partial_i \equiv 0$.*

$$\dots \xrightarrow{\partial_{i+2}} C_{i+1} \xrightarrow{\partial_{i+1}} C_i \xrightarrow{\partial_i} C_{i-1} \xrightarrow{\partial_{i-1}} \dots \xrightarrow{\partial_1} C_0 \xrightarrow{\partial_0} 0$$

**Definition 2.60 (Cycle)** *An $i$-chain $c \in C_i$ is called an **$i$-cycle** if $\partial_i(c) = 0$. The set of all $i$-cycles forms a group called the **$i$-th cycle group** $Z_i = Z_i(X)$ under the addition defined for chains. Moreover, it holds that $Z_i = \ker \partial_i$.*

**Definition 2.61 (Boundary)** *An $i$-chain $c \in C_i$ is called an **$i$-boundary** if there exists $d \in C_{i+1}$, such that $c = \partial_{i+1}(d)$. The set of all $i$-boundaries is a group called the **$i$-th boundary group** $B_i = B_i(X)$. Additionally, it holds that $B_i = \operatorname{Im} \partial_{i+1}$.*

**Example 2.62** *Consider the simplicial complex X in Figure 2.14. Using the boundaries obtained in Example 2.58 we want to analyze the boundaries of the simplices* $[v_0, v_1] + [v_1, v_2] - [v_0, v_2]$ *and* $[v_2, v_3] + [v_3, v_4] - [v_2, v_4]$.

$$\begin{aligned}
\partial_1([v_0, v_1] + [v_1, v_2] - [v_0, v_2]) &= \partial_1([v_0, v_1]) + \partial_1([v_1, v_2]) + \partial_1([v_0, v_2]) = \\
&= v_1 - v_0 + v_2 - v_1 - v_2 + v_0 = \\
&= 0, \\
\partial_1([v_2, v_3] + [v_3, v_4] - [v_2, v_4]) &= \partial_1([v_2, v_3]) + \partial_1([v_3, v_4]) + \partial_1([v_2, v_4]) = \\
&= v_3 - v_2 + v_4 - v_3 - v_4 + v_2 = \\
&= 0.
\end{aligned}$$

*Since both calculations result in* 0*, we could now conclude that these simplices are* 1*-cycles. However, there exists a* 2*-simplex such that*

$$\partial_2([v_2, v_3, v_4]) = [v_3, v_4] - [v_2, v_4] + [v_2, v_3].$$

*Hence, we conclude that the* 1*-simplex* $[v_3, v_4] - [v_2, v_4] + [v_2, v_3]$ *is a boundary and that* $[v_0, v_1] + [v_1, v_2] - [v_0, v_2]$ *is a cycle.*

**Remark 2.63** *The condition* $\partial_{i-1} \circ \partial_i \equiv 0$ *implies that for every i,* $B_i \subseteq Z_i$.

Since the boundary map is a homomorphism, there exist bases that determine its matrix representation.

**Definition 2.64 (Boundary Matrix)** *Let* $\{\sigma_n\}_{n\in\mathbb{N}} \subseteq C_i$ *and* $\{\tau_n\}_{n\in\mathbb{N}} \subseteq C_{i-1}$ *be bases of* $C_i$ *and* $C_{i-1}$ *respectively. The boundary operator* $\partial_i\colon C_i \to C_{i-1}$ *can be represented by a matrix, called the* ***boundary matrix*** $D_i$*, where the columns correspond to the i-simplices and the rows to the* $(i-1)$*-simplices. The* $(n, m)$*-entry of* $D_i$ *is given by*

$$d^i_{nm} = \begin{cases} 1 \text{ if } \sigma_n \in \tau_m, \\ 0 \text{ otherwise.} \end{cases} \tag{2.1}$$

**Example 2.65** *Consider again the simplex in Figure 2.14 for each boundary map. As a reminder, the chain groups of X are given by*

$$\begin{aligned}
C_0(X) &= \{v_0, v_1, v_2, v_3, v_4\}, \\
C_1(X) &= \{[v_0, v_1], [v_0, v_2], [v_1, v_2], [v_2, v_3], [v_2, v_4], [v_3, v_4]\}, \\
C_2(X) &= \{[v_2, v_3, v_4]\},
\end{aligned}$$

*and for* $i > 2, C_i(X) = \{0\}$*. The boundary matrix for every* $i > 1$ *is* $D_i = 0$*, except* *when $i = 0, 1$, then the boundary maps are*

$$D_0 = \begin{array}{c|cccccc|} & v_0v_1 & v_1v_2 & v_0v_2 & v_2v_3 & v_2v_4 & v_3v_4 \\ v_0 & 1 & 0 & 1 & 0 & 0 & 0 \\ v_1 & 1 & 1 & 0 & 0 & 0 & 0 \\ v_2 & 0 & 1 & 1 & 1 & 1 & 0 \\ v_3 & 0 & 0 & 0 & 1 & 0 & 1 \\ v_4 & 0 & 0 & 0 & 0 & 1 & 1 \end{array}, \quad D_1 = \begin{array}{c|c|} & v_2v_3v_4 \\ v_0v_1 & 0 \\ v_1v_2 & 0 \\ v_0v_2 & 0 \\ v_2v_3 & 1 \\ v_2v_4 & 1 \\ v_3v_4 & 1 \end{array}.$$

Everything we proved for the boundary map holds as well for boundary matrices.

**Proposition 2.66** *The matrix product $D_{i-1} \cdot D_i$ is equal to the zero matrix.*

As seen in Example 2.62, an $i$-cycle can also be the boundary of an $(i+1)$-simplex. The idea behind homology is to count the cavities of a topological space by analyzing cycles that are not the boundaries of other simplices.

**Definition 2.67** *Let $k$ be a field, then **$i$-th homology group** $H_i$ of a chain complex $C_\bullet(X)$ with coefficients in $k$ is the group $H_i = Z_i/B_i$. The **$i$-th Betti number** $\beta_i$ is then given by its rank $\beta_i = rank H_i$.*

The most important property of such a construction is its functoriality, which is the reflection of the behavior between simplicial complexes and continuous maps between topological spaces [14].

**Proposition 2.68** *For any abstract simplicial complex $X$ and $Y$ and a continuous map $f \colon X \to Y$ there is an induced linear map $H_n(f) \colon H_n(X) \to H_n(Y)$.*

**Example 2.69** *Here we take into account the complex of Figure 2.14 and the computation of the boundaries in Example 2.58 to compute the homology groups of the chain complex $C_\bullet(X)$. For $i > 2$ we have that $Z_i = B_i = \{0\}$, since in that case $C_i(X) = \{0\}$ therefore $H_i(X) = \{0\}$ for $i > 2$. It remains to calculate $Z_i$ and $B_i$ for $i \leqslant 2$,*

$$\begin{aligned} Z_0 &= \ker \partial_0 = \langle v_0, v_1, v_2, v_3, v_4 \rangle, \\ Z_1 &= \ker \partial_1 = \langle [v_0, v_1] - [v_0, v_2] + [v_1, v_2], [v_2, v_3] + [v_3, v_4] - [v_2, v_4] \rangle, \\ Z_2 &= \ker \partial_2 = \{0\}, \\ B_0 &= \operatorname{Im} \partial_1 = \langle v_0, v_1, v_2, v_3, v_4 \rangle, \\ B_1 &= \operatorname{Im} \partial_2 = \langle [v_2, v_3] + [v_3, v_4] - [v_2, v_4] \rangle \\ B_2 &= \operatorname{Im} \partial_3 = \{0\}. \end{aligned}$$

*After calculating all the groups needed, it remains to take their quotient and get the homology groups $H_i(X)$ for $i \leqslant 2$.*

$$\begin{aligned}
H_0(X) &= Z_0 / B_0 = \langle v_0, v_1, v_2, v_3, v_4 \rangle / \langle v_0, v_1, v_2, v_3, v_4 \rangle \cong \{0\}, \\
H_1(X) &= Z_1 / B_1 = \frac{\langle [v_0, v_1] - [v_0, v_2] + [v_1, v_2], [v_2, v_3] + [v_3, v_4] - [v_2, v_4] \rangle}{\langle [v_2, v_3] + [v_3, v_4] - [v_2, v_4] \rangle} \cong \\
&\cong \langle [v_0, v_1] - [v_0, v_2] + [v_1, v_2] \rangle \cong \mathbb{Z}, \\
H_2(X) &= Z_2 / B_2 = \{0\} / \{0\} = \{0\}.
\end{aligned}$$

*As a last step, we take the ranks of these groups and conclude that*

$$\beta_i = \begin{cases} 1 & \text{if } i = 1, \\ 0 & \text{otherwise,} \end{cases} \qquad H_i(X) = \begin{cases} \mathbb{Z} & \text{if } i = 1; \\ 0 & \text{otherwise.} \end{cases}$$

*So the simplicial complex in Figure 2.14 has only one hole of dimension one. It is connected and it has no* 2*-dimensional holes.*

*From the resulting Betti numbers, we conclude that the simplicial complex in Figure 2.14 is connected, has one hole of dimension one and for all other $i \geqslant 2$ there are no i-dimensional cavities.*

We have now derived simplicial homology, designed to capture the shape of simplicial complexes. Unfortunately, this does not work on every topological space $X$, since the existence of a triangulation of $X$ is not guaranteed. To extend the notion of homology *Eilenberg* defined the more general theory of **singular homology** for any topological space $X$. If $X$ is triangulable, then both the simplicial homology of this space and its singular homology are isomorphic.

## Chapter 3

# Filtrations and Persistent Homology

Understanding the shape of finite metric spaces with homology groups is not helpful, because these do not provide information beyond its number of points. This is the reason for the construction of a different invariant, called *Persistent Homology*. Persistent homology detects holes, shapes and voids of finite metric spaces, after assigning a filtration of simplicial complexes to the point cloud. Most of the theory encountered is based on *Topological Pattern Recognition for Point Cloud Data* [3].

## 3.1 Filtrations

Before we dive into the study of persistent homology, we describe the objects that enable the construction of a topological space from a point cloud [1]. The theory developed in this subchapter is taken from *Computational topology for Data Analysis* [7] and from *Topological Data Analysis with Applications* [4].

To first gain some intuition, we introduce the **Vietoris-Rips complex**, used to construct a simplicial complex, and therefore a topological space, from a point cloud.

**Definition 3.1** *Let $(X, d)$ be a finite metric space. Given a real $r > 0$, the* ***Vietoris-Rips complex*** *is the abstract simplicial complex $VR(X, r)$, with vertex set $X$, and for which $[x_0, \ldots, x_k]$ is a $k$-simplex if and only if $d(x_i, x_j) \leqslant 2r$ for every pair of vertices $0 \leqslant i < j \leqslant k$.*

**Example 3.2** *In Figure 3.1 we have a point cloud on which we construct Vietoris-Rips complexes. Different thresholds $r_i$, lead to simplicial complexes of different homotopy types. For example, in the last simplicial complex, we can notice that a hole appears for the first time.*

---

[1] A finite subset of a metric space $(X, d)$.

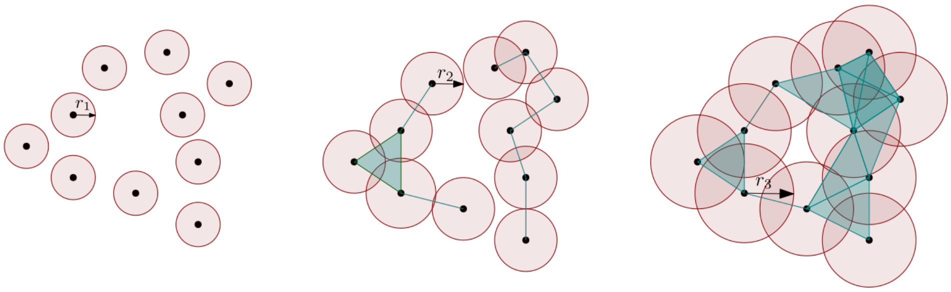


**Figure 3.1:** Examples of Vietoris-Rips complexes for different thresholds.

Taking the collection of all Vietoris-Rips complexes, results in a nested sequence of simplicial complexes $\{VR(X,r)\}_{r\in\mathbb{R}^+}$. For $r \leqslant s$ take a $k$-simplex $\sigma = [x_0,\dots,x_k] \in VR(X,r)$. By definition it holds that $d(x_i,x_j) \leqslant 2r \leqslant 2s$ for every $0 \leqslant i < j \leqslant k$ , thus $\sigma \in VR(X,s)$. This gives a natural inclusion map

$$\iota\colon VR(X,r) \hookrightarrow VR(X,s)$$

for every $r \leqslant s$.

Vietoris-Rips complexes are an example of a *filtration*, which are the objects we introduce next.

**Definition 3.3 (Filtration)** *A **filtration** $\mathcal{F} = \mathcal{F}(X)$ of a topological space $X$ is a nested sequence of topological subspaces $\{X_i\}_{1\leqslant i\leqslant n}$ together with inclusion maps for $i \leqslant j, \iota\colon X_i \hookrightarrow X_j$*

$$\mathcal{F}\colon \emptyset = X_0 \subseteq X_1 \subseteq \cdots \subseteq X_n = X.$$

**Definition 3.4** *Let $K$ be a simplicial complex. A **simplicial filtration** is a nested sequence of subcomplexes of $K$ denoted by $\mathcal{F} = \mathcal{F}(K)$, with*

$$\mathcal{F}\colon \emptyset = K_0 \subseteq K_1 \subseteq \cdots \subseteq K_n = K.$$

*If $K_i \backslash K_{i-1}$ is either empty or a single simplex for $1 \leqslant i \leqslant n$, then $\mathcal{F}$ is called a **simplex-wise** filtration.*

**Example 3.5** *In Figure 3.2 we have an example of a simplicial filtration $\mathcal{F}$ of a simplicial complex $X = (V(X),\Sigma(X))$, with*

$$V(X) = \{a,b,c,d\} \text{ and } \Sigma(X) = \{a,b,c,d,ab,ac,ad,bc,cd,abc\}.$$

*The sequence of subcomplexes $\{X_i\}_{0\leqslant i\leqslant 4}$ is given by*

$$\begin{aligned} X_0 &= \{a,b\}, \quad X_1 = \{a,b,c,d\}, \quad X_2 = \{a,b,c,d,ab,ad,bc\}, \\ X_3 &= \{a,b,c,d,ab,ac,ad,bc\}, \quad X_4 = \{a,b,c,d,ab,ac,ad,bc,cd,abc\}. \end{aligned}$$

*and the inclusion map are $\iota\colon X_i \to X_{i+1}$ for $0 \leqslant i \leqslant 3$.*

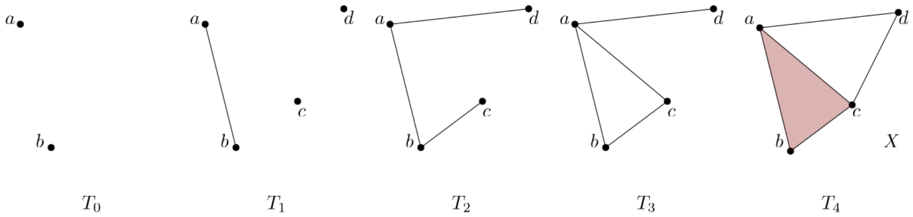


**Figure 3.2:** Example of a simplicial filtration $\mathcal{F}$ of the simplicial complex $X$.

To work with persistent homology, we mostly use a filtration induced by a real-valued function $f\colon X \to \mathbb{R}$ for a topological space $X$.

**Definition 3.6 (Sublevel Sets)** *Let $X$ be a topological space and $f\colon X \to \mathbb{R}$ a function, then the* ***sublevel set of*** *$f$ at $a \in \mathbb{R}$ is defined as:*

$$X_{\leqslant a} = f^{-1}((-\infty, a]) = \{x \in X \mid f(x) \in (-\infty, a]\}.$$

**Remark 3.7** *If not specified, we omit "$\leqslant$" in the subscript:*

$$X_a := X_{\leqslant a} = f^{-1}(-\infty, a].$$

**Definition 3.8 (Sublevel-set Filtration)** *For a topological space $X$ and a real-valued function $f\colon X \to \mathbb{R}$, denote with $X_a = f^{-1}(-\infty, a]$ the sublevel set for the function value $a$. For a sequence of real numbers $a_0 \leqslant a_1 \leqslant \ldots \leqslant a_n$, with $a_0 = -\infty$ and $X_{a_0} = \emptyset$. Then the sequence of subspaces of $X$ connected by inclusions gives the filtration $\mathcal{F}_f$*

$$\mathcal{F}_f\colon \emptyset = X_{a_0} \hookrightarrow X_{a_1} \hookrightarrow X_{a_2} \hookrightarrow \cdots \hookrightarrow X_{a_n}.$$

*called the* ***sublevel-set filtration****.*

**Definition 3.9** *For a topological space $X$ and a function $f\colon X \to \mathbb{R}$ is called* ***tame*** *if the homology groups of its sublevel sets have finite rank and change via inclusion-induced maps only at finitely many points $t \in \mathbb{R}$, called* ***critical points****.*

## 3.2 Persistent Vector Spaces

For a filtration $\mathcal{F}$ of a topological space $X$, whenever $t \leqslant s$ there is an inclusion map $\iota\colon X_t \hookrightarrow X_s$ between subcomplexes $X_t, X_s \subseteq X$. By functoriality these maps induce homomorphisms between homology groups

$$h^i_{t,s} = \iota_*\colon H_i(X_t) \to H_i(X_s)$$

for all $i \geqslant 0$ and $0 \leqslant t \leqslant s$ with $t, s \in \mathbb{R}$. This leads to

$$H_i(\mathcal{F})\colon 0 = H_i(X_0) \to \cdots \to H_i(X_t) \to \cdots \to H_i(X_s) \to \cdots$$

called a **persistent vector space**.

As we noticed in Figures 3.2 and 3.1 the appearing subcomplexes have different homotopy types. Hence, it is important to compute and record homology groups for each of these complexes and see how generators map. The persistent vector space $\{H_i(X_r)\}_{r\in\mathbb{R}}$ contains all this information. To better understand such an object, we introduce the general definition of a persistent vector space and investigate its properties.

**Definition 3.10 (Persistent Vector Space)** *Let $k$ be a field. A family of $k$-vector spaces $\{V_r\}_{r\in\mathbb{R}}$ together with linear transformations $L_V(r,r')\colon V_r \to V_{r'}$, whenever $r \leqslant r'$, is called a* ***persistent vector space*** *if for any $r \leqslant r' \leqslant r''$*

$$L_V(r,r') \cdot L_V(r',r'') = L_V(r,r''). \tag{3.1}$$

Such a construction can be generalized to different objects, like sets, topological spaces, or simplicial complexes. When we speak of a persistent object we mean a family $\{X_r\}_{r\in\mathbb{R}}$ parametrized by $\mathbb{R}$ together with a map $\varphi_X(r,r')\colon X_r \to X_{r'}$ whenever $r \leqslant r'$, with the same property as in Definition 3.1.

**Definition 3.11 (Linear transformation)** *A* ***linear transformation*** *of persistent vector spaces over $k$, $f\colon \{V_r\}_{r\in\mathbb{R}} \to \{W_r\}_{r\in\mathbb{R}}$, is a family of linear transformations $f_r\colon V_r \to W_r$, such that for all $r \leqslant r'$,*

$$f_{r'} \circ L_V(r,r') = L_W(r,r') \circ f_r.$$

*This is equivalent to the following diagram being commutative*

$$\begin{tikzcd}
V_r \arrow[r, "L_V(r{,}r')"] \arrow[d, "f_r"'] & V_{r'} \arrow[d, "f_{r'}"] \\
W_r \arrow[r, "L_W(r{,}r')"] & W_{r'}.
\end{tikzcd}$$

*We call such a linear transformation an* ***isomorphism*** *if it admits a two-sided inverse.*

**Example 3.12** *Consider the filtration $\mathcal{F}$ from Figure 3.2. We can see that the $0$-simplices do not appear all at once: $a$ and $b$ appear at time $T_0$, and $c$, $d$ at time $T_1$. Hence, the persistent vector space for the $0$-simplices is given by*

$$(C_0(\mathcal{F}))_r = \begin{cases} \langle a,b\rangle, & r\in[T_0,T_1), \\ \langle a,b,c,d\rangle, & r\in[T_1,\infty). \end{cases}$$

*The persistent vector space for the* 1*-simplices is given by*

$$(C_1(\mathcal{F}))_r = \begin{cases} \{0\} & r \in [T_0, T_1), \\ \langle ab \rangle, & r \in [T_1, T_2), \\ \langle ab, ad, bc \rangle, & r \in [T_2, T_3), \\ \langle ab, ac, ad, bc \rangle & r \in [T_3, T_4), \\ \langle ab, ac, ad, bc, cd \rangle & r \in [T_4, \infty). \end{cases}$$

*Lastly,*

$$(C_2(\mathcal{F}))_r = \begin{cases} \{0\}, & r \in [T_0, T_4), \\ \langle abc \rangle, & r \in [T_4, \infty). \end{cases}$$

*For $k > 2$ we have that $(C_k(\mathcal{F}))_r = \{0\}$ for every $r > 0$ since there are no simplices of dimension bigger than* 2*. For these persistent vector spaces, we can additionally define linear maps*

$$\partial_i \colon \{(C_i(\mathcal{F}))_r\}_{r\in\mathbb{R}^+} \to \{(C_{i-1}(\mathcal{F}))_r\}_{r\in\mathbb{R}^+}.$$

*Note that after fixing $r \in \mathbb{R}^+$, these maps can be interpreted as maps between vector spaces, corresponding to the notion of boundary maps from homology theory.*

**Definition 3.13 (Sub-persistent Vector Space)** *A **sub-persistent vector space** of $\{V_r\}_{r\in\mathbb{R}}$ is a choice of $k$-subspaces $U_r \subseteq V_r$ for all $r \in [0, +\infty)$, such that $L_V(r, r')(U_r) \subseteq U_{r'}$ for all $r \leqslant r'$.*

**Remark 3.14** *If $f \colon \{V_r\}_{r\in\mathbb{R}} \to \{W_r\}_{r\in\mathbb{R}}$ is a linear transformation, then $im(f)$ is the sub-persistent vector space $\{im(f_r)\}_{r\in\mathbb{R}}$.*

In our work, we mostly focus on persistent vector spaces with parameter $r \in [0, +\infty)$. Whenever $r < 0$ we set $V_r = \{0\}$.

**Definition 3.15 (Quotient Persistent Vector Space)** *If $\{U_r\}_{r\in\mathbb{R}} \subseteq \{V_r\}_{r\in\mathbb{R}}$ is a sub-persistent vector space, the persistent vector space $\{V_r/U_r\}_{r\in\mathbb{R}}$, where the linear transformation $L_{V/U}(r, r')$ for $r \leqslant r'$ is given by*

$$\begin{aligned} V_r/U_r &\to V_{r'}/U_{r'} \\ [v] &\mapsto [L_V(r, r')(v)] \end{aligned}$$

*is called the **quotient persistent vector space**.*

**Definition 3.16 ($\mathbb{R}^+$-filtered set)** *A $\mathbb{R}^+$**-filtered set** $(X, \rho)$ is a set $X$ equipped with a function $\rho \colon X \to [0, \infty)$.*

**Definition 3.17 (Free $k$-vector space)** *Let $k$ be a field and $X$ a finite set, then the **free** $k$**-vector space on the set** $X$, denoted by $V_k(X)$ is the $k$-linear span of the set $X$.*

**Definition 3.18 (Free Persistent Vector Space)** *Let $(X, \rho)$ be a $\mathbb{R}^+$-filtered set. A **free persistent vector space** on $(X, \rho)$ is the persistent vector space denoted by $\{V_k(X, \rho)_r\}_{r \in \mathbb{R}}$ such that $V_k(X, \rho)_r \subseteq V_k(X)$, where $V_k(X, \rho)_r$ is the k-linear span of the set $X[r] \subseteq X$ defined by $X[r] = \{x \in X | \rho(x) \leqslant r\}$.*

**Remark 3.19** *If $r \leqslant r'$ it holds $X[r] \subseteq X[r']$, that implies the inclusion*

$$V_k(X, \rho)_r \subseteq V_k(X, \rho)_{r'}.$$

**Definition 3.20** *A persistent vector space $\{W_r\}_{r \in \mathbb{R}}$ is called **free** if there exists a $\mathbb{R}^+$-filtered set $(X, \rho)$, such that $\{W_r\}_{r \in \mathbb{R}} \cong \{V_k(X, \rho)_r\}_{r \in \mathbb{R}}$. Moreover, $\{W_r\}_{r \in \mathbb{R}}$ is called **finitely generated**, if $X$ is finite.*

**Proposition 3.21** *A linear combination $\sum_x a_x x \in V_k(X)$ lies in $V_k(X, \rho)_r$ if and only if $a_x = 0$ for all $x \in X$ with $\rho(x) > r$.*

**Proof** Let $v = \sum_{x \in X} a_x x \in V_k(X)$.
"$\Rightarrow$" Assume that $v \in V_k(X, \rho)_r$ for a $r > 0$, then by definition $v$ is a linear combination of $x \in X$ such that $\rho(x) \leqslant r$, hence $v = \sum_{\substack{x \in X \\ \rho(x) \leqslant r}} a_x x$, thus by linear independence it immediately follows that for $x \in X$ with $\rho(x) > r, a_x = 0$.
"$\Leftarrow$" Assume that $a_x = 0$ for all $x \in X$ with $\rho(x) > r$, then

$$v = \sum_{\substack{x \in X \\ \rho(x) \leqslant r}} a_x x + \underbrace{\sum_{\substack{x \in X \\ \rho(x) > r}} a_x x}_{=0} = \sum_{\substack{x \in X \\ \rho(x) \leqslant r}} a_x x$$

by assumption, therefore $v \in V_k(X, \rho)_r$. □

**Definition 3.22 (Finitely Presented Persistence Vector Space)** *A persistent vector space $\{V_r\}_{r \in \mathbb{R}}$ is called **finitely presented** if there exists a linear transformation $f \colon \{V_r\}_{r \in \mathbb{R}} \to \{W_r\}_{r \in \mathbb{R}}$, such that*

$$\{V_r\}_{r \in \mathbb{R}} \cong \{W_r\}_{r \in \mathbb{R}} / Im(f)$$

*and both $\{V_r\}_{r \in \mathbb{R}}$, $\{W_r\}_{r \in \mathbb{R}}$ are finitely generated free persistent vector spaces.*

From linear algebra, it is known that if we choose a basis for the vector spaces $V$ and $W$, every linear transformation $f \colon V \to W$ has a corresponding matrix representation. A similar procedure also exists for persistent vector spaces.

**Definition 3.23** *Let $(X, Y)$ be a pair of finite sets and k a field. $A(X, Y)$**-matrix** is an array $[a_{xy}]$ of elements $a_{xy} \in k$ for $x \in X, y \in Y$. We denote by $r(x)$ the row corresponding to $x \in X$ and by $c(y)$ the column corresponding to $y \in Y$.*

**Remark 3.24** *There exists an $r$ large enough such that $\{V_k(X, \rho)_r\} = V_k(X)$ since $X$ is finite. Hence, for any linear map $f \colon \{V_k(X, \rho)_r\}_{r \in \mathbb{R}} \to \{V_k(Y, \sigma)_r\}_{r \in \mathbb{R}}$ between finitely generated free persistent vector spaces there exists a map*

$$f_\infty \colon V_k(X) \to V_k(Y)$$

*between finite-dimensional vector spaces over k. Therefore, the existence of bases* $\{x\}_{x\in X} \subseteq V_k(X)$ *and* $\{y\}_{y\in Y} \subseteq V_k(Y)$ *is guaranteed. The map f between persistent vector spaces is represented by the* $(X,Y)$*-matrix* $A(f) = [a_{xy}]$ *with* $a_{xy} \in k$.

**Proposition 3.25** *The* $(X,Y)$*-matrix* $A(f)$ *has the property that*

$$a_{xy} = 0 \quad \textit{if} \quad \rho(x) > \sigma(y). \tag{3.2}$$

*Any* $(X,Y)$*- matrix A satisfying Equation 3.2 uniquely determines a linear transformation between persistent vector spaces*

$$f_A \colon \{V_k(Y,\sigma)_r\}_{r\in\mathbb{R}} \to \{V_k(X,\rho)_r\}_{r\in\mathbb{R}}$$

*and the correspondence* $f \to A(f)$ *and* $A \to f_A$ *are inverses to each other.*

**Proof** Take $y \in Y$ a basis vector, then since $y$ is a generator of $V_k(Y,\sigma)_{\sigma(y)}$ it holds that $y \in V_k(Y,\sigma)_{\sigma(y)}$. On the other hand, by definition of $f$

$$f(y) = \sum_{x\in X} a_{xy}x$$

is a linear combination of basis vectors of $V_k(X,\rho)_{\sigma(y)}$. By Proposition 3.21 such a linear combination lies in $V_k(X,\rho)_{\sigma(y)}$ if and only if $a_{xy} = 0$ whenever $\rho(x) > \sigma(y)$. ☐

The matrices we analyzed in the last proposition have a specific name and are important in the study of persistent homology.

**Definition 3.26** *Let* $(X,\rho)$ *and* $(Y,\sigma)$ *be two* $\mathbb{R}^+$*-filtered sets. An* $(X,Y)$*- matrix* $A = [a_{xy}]$ *satisfying the condition that* $a_{xy} = 0$*, whenever* $\rho(x) > \sigma(y)$ *is called* $(\rho,\sigma)$***- adapted***.

Moreover, for $\mathbb{R}^+$-filtered sets $(X,\rho)$ and $(Y,\sigma)$ with maps $\rho$ and $\sigma$ both $[0,\infty)$-valued, any $(\rho,\sigma)$-adapted matrix $A = [a_{xy}]$ determines a persistent vector space via the map

$$\theta\colon A \to \{(V_k(X,\rho)/\operatorname{Im}(f_A))_r\}_{r\in\mathbb{R}^+}, \tag{3.3}$$

where $f_A\colon \{V_k(Y,\sigma)_r\}_{r\in\mathbb{R}} \to \{V_k(X,\rho)_r\}_{r\in\mathbb{R}}$ is the uniquely determined linear transformation defined in Proposition 3.25 between persistent vector spaces. For such a matrix $A$ the space $\theta(A)$ always defines a finitely presented persistent vector space.

An example of a finitely presented persistent vector space is given by the *interval persistent vector spaces.*

**Definition 3.27** *An* ***interval persistent vector space*** $P(a,b)$ *for a pair* $(a,b)$ *with* $a \in \mathbb{R}_+, b \in \mathbb{R}_+ \cup \{+\infty\}$ *and* $a < b$ *is defined as*

$$P(a,b)_r = \begin{cases} k & \textit{if} \quad r \in [a,b) \\ 0 & \textit{if} \quad r \notin [a,b) \end{cases},$$

*where k is a field. We define the linear map for $r < r'$*

$$L_{P(a,b)}(r,r') = \begin{cases} id_k & \text{if} \quad r, r' \in [a,b), \\ 0 & \text{else}. \end{cases}$$

Observe the $\mathbb{R}^+$-filtered sets $(X,\rho)$ and $(Y,\sigma)$, then we can derive the corresponding $(\rho,\sigma)$-adapted matrices, which depend on the value taken by $b$ and derive the persistent vector space.

- $b = +\infty$: From the definition above one can capture that for $a \in \mathbb{R}^+$ $P(a,b)_r = k$, since then $r \in [a,b)$. Moreover $P(a,b)$ is finitely presented and $P(a,b) \cong V_k(X,\rho)$. By looking at the map given in Equation 3.3 one can note that $A$ has to be the zero map, henceforth $P(a,b) \cong \theta([0])$.
- $b \in \mathbb{R}^+$ *is finite*: Let $X = \{x\}$ and $Y = \{y\}$ contain one single element and $\rho(x) = a$ and $\sigma(y) = b$. Then the $(X,Y)$-matrix [1] is a one-dimensional $(\rho,\sigma)$-adapted matrix, since $a \leqslant b$. From these facts, $P(a,b) \cong \theta([1])$.

**Example 3.28** *In Example 3.12 we mentioned the existence of boundary maps*

$$\partial_i \colon \{C_i(\mathcal{F})_r\}_{r\in\mathbb{R}} \to \{C_{i-1}(\mathcal{F})_r\}_{r\in\mathbb{R}}.$$

*It is now possible to write down the boundary matrices that record at "which time" each simplex appears. For example, denoting the function $\rho \colon \Sigma(X) \to \mathbb{R}^+$ and taking the simplex bc, we notice that above the second column, we have $(bc,2)$, which means that the 1-simplex bc appears ar time $\rho(bc) = 2$.*

$$(D_0)_\infty = \begin{array}{c|ccccc|} & (ab,1) & (bc,2) & (ad,2) & (ac,3) & (cd,4) \\ \hline (d,1) & 0 & 0 & 1 & 0 & 1 \\ (c,1) & 0 & 1 & 0 & 1 & 1 \\ (b,0) & 1 & 1 & 0 & 0 & 0 \\ (a,0) & 1 & 0 & 1 & 1 & 0 \\ \hline \end{array}, \quad (D_1)_\infty = \begin{array}{c|c|} & (abc,4)) \\ \hline (cd,4) & 0 \\ (ac,3) & 1 \\ (ad,2) & 0 \\ (bc,2) & 1 \\ (ab,1) & 1 \\ \hline \end{array}.$$

## 3.3 Decomposition Theorem of Persistent Vector Spaces

In the following, we state three important propositions needed to prove the representation theorem for finitely persistent vector spaces.

**Proposition 3.29** *For any finitely presented persistent vector space $\{V_r\}_{r\in\mathbb{R}}$ there exists a matrix $A$, such that $\{V_r\}_{r\in\mathbb{R}} \cong \theta(A)$.*

**Proposition 3.30** *Let $(X,\rho)$ be an $\mathbb{R}^+$- filtered set. Then the group $Aut(V_k(X,\rho))$ is identified with the group of all $(\rho,\rho)$-adapted $(X,X)$-matrices under the correspondence between matrices and linear transformations given in proposition 3.25.*

These propositions are a direct consequence of the correspondence between matrices and linear maps stated in 3.25.

**Proposition 3.31** *Let $(X,\rho)$ and $(Y,\sigma)$ be $\mathbb{R}^+$-filtered sets, and $A$ a $(\rho,\sigma)$-adapted $(X,Y)$-matrix. Let now $B$ be a $(\rho,\rho)$-adapted $(X,X)$- matrix and $C$ a $(\sigma,\sigma)$-adapted $(Y,Y)$-matrix. Then $BAC$ is also $(\rho,\sigma)$-adapted $(X,Y)$-matrix, and there is an isomorphism between the persistent vector spaces $\theta(A) \cong \theta(BAC)$.*

To compute kernels and images in linear algebra entails working with Gaussian operations. For the $(\rho,\sigma)$-adapted matrices to keep their properties, we need to introduce some specific operations.

**Definition 3.32** *Let $(X,\rho),(Y,\sigma)$ be two $\mathbb{R}^+$-filtered sets.*

- *An **adapted row operation** is an operation that adds a multiple of $r(x)$ to $r(x')$, whenever $\rho(x) \geqslant \rho(x')$.*
- *An **adapted column operation** is an operation that adds a multiple of $c(y)$ to $c(y')$, when $\sigma(y) \leqslant \sigma(y')$.*

Additionally to adapted row and column operations, we can

- permute columns or rows,
- multiply columns or rows by a scalar $\alpha \in K$ with $\alpha \neq 0$,

independently of the $\rho$- and $\sigma$-values.

Now we have all the needed tools to prove the representation theorem. We state the theorem in two parts. First, we state and prove the existence of such a decomposition.

**Theorem 3.33** *Let $k$ be a field, then every finitely presented persistent vector space $\{V_r\}_{r\in\mathbb{R}}$ over $k$ is isomorphic to a finite direct sum*

$$\{V_r\}_{r\in\mathbb{R}} = \bigoplus_{i=1}^{n} P(a_i,b_i) = P(a_1,b_1) \oplus \ldots \oplus P(a_n,b_n)$$

*for $a_i \in \mathbb{R}^+$ and $b_i \in \mathbb{R}^+ \cup \{+\infty\}$ for all $i \in \{1,\ldots,n\}$.*

**Proof** From Proposition 3.29 we know that for every finitely presented persistent vector space $\{V_r\}_{r\in\mathbb{R}}$ there exists a $(\rho,\sigma)$-adapted $(X,Y)$-matrix $A$ such that $\{V_r\}_{r\in\mathbb{R}} \cong \theta(A)$. First assume w.l.o.g that $A$ is a matrix with at most one non-zero entry in each row and column, and that the non-zero entry is equal to one. This means that up to permutations of rows and columns, $A$ has the form

$$A = \begin{bmatrix} I_n & 0_{n\times k} \\ 0_{m\times n} & 0_{m\times k} \end{bmatrix}. \tag{3.4}$$

Take $\{(x_1, y_1), \dots, (x_n, y_n)\}$ to be the set of all pairs such that $a_{x_i y_i} = 1$ for every $1 \leqslant i \leqslant n$. From the correspondence between linear maps and matrices given in Proposition 3.25 and the map $\theta$ in Equation 3.3 it holds:

$$\begin{aligned}\theta(A) \cong V_k(X, \rho) &= \bigoplus_{x \in X} V_k(x, \rho) = \\ &= \bigoplus_{i=1}^{n} V_k(x, \rho) \oplus \bigoplus_{x \in X \setminus \{x_1, \dots, x_n\}} V_k(x, \rho) = \\ &= \bigoplus_{i=1}^{n} P(\rho(x_i), \sigma(y_1)) \oplus \bigoplus_{x \in X \setminus \{x_1, \dots, x_n\}} P(\rho(x), +\infty).\end{aligned}$$

The last equality holds because of the identity of interval persistent vector spaces stated in Remark 3.2.

It now remains to show that we can modify every $(\rho, \sigma)$-adapted $(X, Y)$-matrix $A$ to reach the form in Equation 3.4.

Since $X$ and $Y$ are both finite, we can find $y \in Y$, such that

$$\sigma(y) = \min\{\sigma(y') \mid y' \in Y, c(y') \neq 0\}$$

and $x \in X$ with

$$\rho(x) = \max_{x' \in X} \rho(x')$$

and $a_{xy} \neq 0$. Now we can apply adapted row and column operations.
Since $x$ is chosen to have the maximal $\rho$-value, we can add $r(x)$ to every other $r(x')$ for $x, x' \in X$. Similarly, $y$ is chosen to have minimal $\sigma$-value, therefore we can add $c(y)$ to every other $c(y')$ for $y, y' \in Y$.

1. Add $r(x)$ to $r(x')$ until each entry of the column $c(y)$ is zero except $a_{xy}$.
2. Add $c(y)$ to $c(y')$ until each entry of the row $r(x)$ is zero except $a_{xy}$.

The above algorithm gives a matrix, where the only non-zero element of $r(x)$ and $c(y)$ is $a_{xy}$. To conclude the algorithm, we multiply $r(x)$ by $\frac{1}{a_{xy}}$.

We now proceed inductively and do the same operations on the $(\rho', \sigma')$-adapted $(X \setminus \{x\}, Y \setminus \{y\})$ matrix with removed $r(x)$ and $c(y)$, where $\rho'$ and $\sigma'$ are restrictions of $\rho$ and $\sigma$ to $(X \setminus \{x\}), (Y \setminus \{y\})$ respectively. This process does not affect $r(x)$ and $c(y)$.
The adapted row and column operations we applied correspond to multiplication on the left with a $(\rho, \rho)$-adapted $(X, X)$-matrix $B$ and on the right with a $(\sigma, \sigma)$-adapted $(Y, Y)$-matrix $C$, from which then $BAC$ has the desired properties and from Proposition 3.31 we get that $\theta(A) \cong \theta(BAC)$ and have therefore the desired result. ☐

As a second step, we state and prove that such a decomposition is unique.

**Theorem 3.34** *Let $\{V_r\}_{r\in\mathbb{R}}$ be a finitely presented persistent vector space over a field k and assume that there exist two decompositions*

$$\{V_r\}_{r\in\mathbb{R}} \cong \bigoplus_{i\in I} P(a_i, b_i) \quad \text{and} \quad \{V_r\}_{r\in\mathbb{R}} \cong \bigoplus_{j\in J} P(c_j, d_j)$$

*where I and J are finite sets. Then $|I| = |J|$ and the set of pairs $\{(a_i, b_i)\}_{i\in I}$ occurring is identical to the multiset of pairs $\{(c_j, d_j)\}_{j\in J}$.*

**Proof** Let $a_{\min} = \min\limits_{1\leqslant i\leqslant n} a_i$ and $c_{\min} = \min\limits_{1\leqslant j\leqslant n} c_j$ denote the minimal value over all $a_i$ and $c_j$ respectively. At the same time, both minima are represented by $\{r \in \mathbb{R} \mid V_r \neq 0\}$, therefore $a_{\min} = c_{\min}$. Next define

$$b_{\min} = \min\{b_i \mid a_i = a_{\min}\} \text{ and } d_{\min} = \min\{d_j \mid c_j = c_{\min}\}$$

both are again naturally characterized through $\min\{r' \mid \ker(L(r, r')) \neq \{0\}\}$, so $b_{\min} = d_{\min}$. From these two equalities we see that $P(a_{\min}, b_{\min}) = P(c_{\min}, d_{\min})$ and this interval appears in both decompositions. Now we want to analyze how many times $P(a_{\min}, b_{\min})$ occurs in each decomposition. Consider the sum of all occurrences of the summand $P(a_{\min}, b_{\min})$ in both decompositions. Since these sums are sub-persistent vector spaces of $\{V_r\}_{r\in\mathbb{R}}$, there is a characterization of both sums as the sub-persistent vector space $\{W_r\}_{r\in\mathbb{R}}$ given by the kernel of the linear map

$$L(r, b_{\min})|_{\operatorname{Im}(L(a_{\min}, r))} \colon \operatorname{Im}(L(a_{\min}, r)) \longrightarrow V_{b_{\min}}.$$

From this characterization the number of summands of the form $P(a_{\min}, b_{\min})$ in both decompositions is the same. Specifically, denote by

$$I' = \{i \in I \mid a_i = a_{\min}, b_i = b_{\min}\} \text{ and } J' = \{j \in J \mid c_j = c_{\min}, d_j = d_{\min}\}.$$

Then it holds $|I'| = |J'|$. Forming the quotient of $\{V_r\}_{r\in\mathbb{R}}$ by $\{W_r\}_{r\in\mathbb{R}}$ we get the identification

$$\{V_r\}_{r\in\mathbb{R}} / \{W_r\}_{r\in\mathbb{R}} \cong \bigoplus_{i\in I\setminus I'} P(a_i, b_i) \text{ and } \{V_r\}_{r\in\mathbb{R}} / \{W_r\}_{r\in\mathbb{R}} \cong \bigoplus_{j\in J\setminus J'} P(c_j, d_j).$$

By inductively repeating this procedure on the number of summands we obtain uniqueness of the decomposition. □

In the beginning of Section 3.2, we saw that for a simplicial complex $X$ with subcomplexes $\{X_r\}_r$ and a corresponding simplicial filtration $\mathcal{F}$ the collection $\{H_i(X_r)\}_{r\in\mathbb{R}}$ forms a persistent vector space. From Theorem 3.33 we deduce that for this vector space there exists a unique decomposition in a direct sum of interval persistent vector spaces

$$\{H_i(X_r)\}_{r\in\mathbb{R}} \cong \bigoplus_{j=1}^{n} P(a_j, b_j)$$

for $i \geqslant 0$ and some $a_j, b_j \in \mathbb{R}$ for each $1 \leqslant j \leqslant n$. From this decomposition we gain a more visual representation of what homology groups represent. The above direct sum means that for each dimension $i \geqslant 0$ there are n $i$-dimensional holes that get born at time $a_j$ and die at $b_j$.

There is an algorithm for computing the homology of such spaces by using adapted row and column operations instead of the usual arbitrary operations taught in linear algebra. Thanks to this algorithm we can produce a presentation for persistent homology.

### 3.3.1 Persistent Homology Computation

In this section we work with $\mathbb{Z}_2$-coefficients, since it simplifies the computation and the main results don't change. Recall also that for boundary matrices $D_i$, $D_{i-1}$ for $i \geqslant 1$ it always holds that $D_{i-1} \cdot D_i = 0$.

Before we dive into the computation of persistent homology groups, we first need to study some properties for pairs of matrices $(A, B)$, with $A \cdot B = 0$. We work with $\mathbb{R}^+$-filtered sets $(X, \rho), (Y, \sigma), (Z, \tau)$, where $A$ is a $(\rho, \sigma)$-adapted $(X, Y)$-matrix and $B$ is a $(\sigma, \tau)$-adapted $(Y, Z)$-matrix. Recall that working with a pair of matrices $(A, B)$ requires some alterations of the Gaussian operations.

- Arbitrary adapted row operations on $A$.
- Arbitrary adapted column operations on $B$.
- Adapted column operations on $A$ have to be done simultaneously with adapted row operations on $B$ in the following way:
  - Multiplication of the $i$-th column of $A$ by $\alpha \neq 0$ corresponds to multiplication of the $i$-th row of $B$ by $\alpha^{-1}$.
  - Permutation of two columns of $A$ corresponds to the permutation of two rows of $B$.
  - Addition of the $i$-th column times $\beta$ to the $j$-th column of $A$ corresponds to adding $\beta$ times the $j$-th row to the $i$-th row of $B$.

Applying any of these operations to a pair of matrices $(A, B)$ guarantees that the resulting pair $(A', B')$ also fulfills the property $A' \cdot B' = 0$.

**Remark 3.35** *For a matrix $A$ over a field k, there is always a sequence of row and column operations that lead $A$ to a matrix of the form $\begin{bmatrix} I_n & 0 \\ 0 & 0 \end{bmatrix}$, where $n = \text{rank } A$.*

**Proposition 3.36** *Let $(A, B)$ be a pair of matrices with $A \cdot B = 0$. There exists a sequence of the operations described above that leads to $(A', B')$ with*

$$\left( \begin{bmatrix} I_n & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}, \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & I_m \end{bmatrix} \right). \tag{3.5}$$

*The pair $(A', B')$ is uniquely determined by $(A, B)$ and it holds that $n = rankA$, $m = rankB$.*

**Remark 3.37** *Proposition 3.36 also holds for $(\rho, \sigma)$-adapted $(X, Y)$-matrices and $(\sigma, \tau)$-adapted $(Y, Z)$-matrices, where $(X, \rho), (Y, \sigma), (Z, \tau)$ are $\mathbb{R}^+$-filtered sets.*

To compute persistent homology groups we simultaneously apply adapted row and column operations on a pair of boundary matrices $(D_{i-1}, D_i)$, for $i \geqslant 1$ to reach the form in Equation 3.5. In this way, we are able to read off $\ker(D_{i-1})$ and $\mathrm{Im}(D_i)$, from which one can compute the $i$-th homology group and the corresponding Betti number.

**Remark 3.38** *For convenience, we order the columns in increasing order of the $\sigma$-values and the rows in descending order of $\rho$-values.*

Let us now demonstrate this procedure on an example.

**Example 3.39** *We now compute the 1-dimensional homology group of the filtration $\mathcal{F}$ of the simplicial complex $X$ illustrated in Figure 3.2. Recall that the boundary matrices are given by*

$$(D_0)_\infty = \begin{array}{c} \\ (d,1) \\ (c,1) \\ (b,0) \\ (a,0) \end{array} \begin{array}{c} \begin{array}{ccccc} (ab,1) & (bc,2) & (ad,2) & (ac,3) & (cd,4) \end{array} \\ \left[ \begin{array}{ccccc} 0 & 0 & 1 & 0 & 1 \\ 0 & 1 & 0 & 1 & 1 \\ 1 & 1 & 0 & 0 & 0 \\ 1 & 0 & 1 & 1 & 0 \end{array} \right] \end{array}, (D_1)_\infty = \begin{array}{c} \\ (cd,4) \\ (ac,3) \\ (ad,2) \\ (bc,2) \\ (ab,1) \end{array} \begin{array}{c} (abc,4)) \\ \left[ \begin{array}{c} 0 \\ 1 \\ 0 \\ 1 \\ 1 \end{array} \right] \end{array}.$$

*and that we can only apply adapted row and column operations. Our goal is to transform the matrix pair $(D_0, D_1)$ into a matrix of the form as in Equation 3.5.*

*We begin by modifying $a_{1,1}$ from 0 to 1 and making sure that this is the only non-zero entry in $r_1$ and $c_1$. To do so, we first switch the columns $c_1$ and $c_2$. Secondly, we eliminate all the non-zero entries in $r_1$ and $c_1$, leaving only $a_{1,1} = 1$. In this case, we need to pay attention to the arising time of the simplices, since we switched the first and second columns. Note that we only denote the operations done on the matrix on* *the left. However, we apply the corresponding operation on the matrix on the right.*

$$\left(\begin{array}{cc} & \begin{array}{ccccc} (ab,1) & (ad,2) & (bc,2) & (ac,3) & (cd,4) \end{array} \\ \begin{array}{c} (1) \\ (1) \\ (0) \\ (0) \end{array} & \left[\begin{array}{ccccc} 0 & 1 & 0 & 0 & 1 \\ 0 & 0 & 1 & 1 & 1 \\ 1 & 0 & 1 & 0 & 0 \\ 1 & 1 & 0 & 1 & 0 \end{array}\right] \end{array}, \begin{array}{cc} & (abc,4) \\ \begin{array}{c} (ab,1) \\ (ad,2) \\ (bc,2) \\ (ac,3) \\ (cd,4) \end{array} & \left[\begin{array}{c} 1 \\ 0 \\ 1 \\ 1 \\ 0 \end{array}\right] \end{array}\right)$$

$$\xrightarrow{(c_1 \leftrightarrow c_2)}$$

$$\left(\begin{array}{cc} & \begin{array}{ccccc} (ad,2) & (ab,1) & (bc,2) & (ac,3) & (cd,4) \end{array} \\ \begin{array}{c} (1) \\ (1) \\ (0) \\ (0) \end{array} & \left[\begin{array}{ccccc} 1 & 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 1 & 1 \\ 0 & 1 & 1 & 0 & 0 \\ 1 & 1 & 0 & 1 & 0 \end{array}\right] \end{array}, \begin{array}{cc} & (abc,4) \\ \begin{array}{c} (ad,2) \\ (ab,1) \\ (bc,2) \\ (ac,3) \\ (cd,4) \end{array} & \left[\begin{array}{c} 0 \\ 1 \\ 1 \\ 1 \\ 0 \end{array}\right] \end{array}\right)$$

$$\xrightarrow{(c_1 + c_5 \rightarrow c_5)}$$

$$\left(\begin{array}{cc} & \begin{array}{ccccc} (ad,2) & (ab,1) & (bc,2) & (ac,3) & (ad{+}cd,4) \end{array} \\ \begin{array}{c} (1) \\ (1) \\ (0) \\ (0) \end{array} & \left[\begin{array}{ccccc} 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 1 & 1 \\ 0 & 1 & 1 & 0 & 0 \\ 1 & 1 & 0 & 1 & 1 \end{array}\right] \end{array}, \begin{array}{cc} & (abc,4) \\ \begin{array}{c} (ad,2) \\ (ab,1) \\ (bc,2) \\ (ac,3) \\ (ad{+}cd,4) \end{array} & \left[\begin{array}{c} 0 \\ 1 \\ 1 \\ 1 \\ 0 \end{array}\right] \end{array}\right)$$

$$\xrightarrow{(r_1 + r_4 \rightarrow r_4)}$$

$$\left(\begin{array}{cc} & \begin{array}{ccccc} (ad,2) & (ab,1) & (bc,2) & (ac,3) & (ad{+}cd,4) \end{array} \\ \begin{array}{c} (1) \\ (1) \\ (0) \\ (0) \end{array} & \left[\begin{array}{ccccc} 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 1 & 1 \\ 0 & 1 & 1 & 0 & 0 \\ 0 & 1 & 0 & 1 & 1 \end{array}\right] \end{array}, \begin{array}{cc} & (abc,4) \\ \begin{array}{c} (ad,2) \\ (ab,1) \\ (bc,2) \\ (ac,3) \\ (ad{+}cd,4) \end{array} & \left[\begin{array}{c} 0 \\ 1 \\ 1 \\ 1 \\ 0 \end{array}\right] \end{array}\right)$$

*As one might notice, the left matrix has now only* $a_{1,1}$ *as a non-zero entry in row* $r_1$ *and column* $c_1$*, which is exactly what we wanted to achieve. At this point, we do a* *similar procedure on the submatrix on the left until we reach a form with non-zero entries only on the diagonal.*

$$\left(\begin{array}{c|ccccc|} & (ad,2) & (ab,1) & (bc,2) & (ac,3) & (ad+cd,4) \\ (1) & 1 & 0 & 0 & 0 & 0 \\ (1) & 0 & 0 & 1 & 1 & 1 \\ (0) & 0 & 1 & 1 & 0 & 0 \\ (0) & 0 & 1 & 0 & 1 & 1 \end{array}\;,\; \begin{array}{c|c|} & (abc,4) \\ (ad,2) & 0 \\ (ab,1) & 1 \\ (bc,2) & 1 \\ (ac,3) & 1 \\ (ad+cd,4) & 0 \end{array}\right)$$

$$\xrightarrow{(c_2 \leftrightarrow c_3)}$$

$$\left(\begin{array}{c|ccccc|} & (ad,2) & (bc,2) & (ab,1) & (ac,3) & (ad+cd,4) \\ (1) & 1 & 0 & 0 & 0 & 0 \\ (1) & 0 & 1 & 0 & 1 & 1 \\ (0) & 0 & 1 & 1 & 0 & 0 \\ (0) & 0 & 0 & 1 & 1 & 1 \end{array}\;,\; \begin{array}{c|c|} & (abc,4) \\ (ad,2) & 0 \\ (bc,2) & 1 \\ (ab,1) & 1 \\ (ac,3) & 1 \\ (ad+cd,4) & 0 \end{array}\right)$$

$$\xrightarrow{(r_2 + r_3 \to r_3)}$$

$$\left(\begin{array}{c|ccccc|} & (ad,2) & (bc,2) & (ab,1) & (ac,3) & (ad+cd,4) \\ (1) & 1 & 0 & 0 & 0 & 0 \\ (1) & 0 & 1 & 0 & 1 & 1 \\ (0) & 0 & 0 & 1 & 1 & 1 \\ (0) & 0 & 0 & 1 & 1 & 1 \end{array}\;,\; \begin{array}{c|c|} & (abc,4) \\ (ad,2) & 0 \\ (bc,2) & 1 \\ (ab,1) & 1 \\ (ac,3) & 1 \\ (ad+cd,4) & 0 \end{array}\right)$$

$$\xrightarrow{(c_2 + c_4 \to c_4)}$$

$$\left(\begin{array}{c|ccccc|} & (ad,2) & (bc,2) & (ab,1) & (ac+bc,3) & (ad+cd,4) \\ (1) & 1 & 0 & 0 & 0 & 0 \\ (1) & 0 & 1 & 0 & 0 & 1 \\ (0) & 0 & 0 & 1 & 1 & 1 \\ (0) & 0 & 0 & 1 & 1 & 1 \end{array}\;,\; \begin{array}{c|c|} & (abc,4) \\ (ad,2) & 0 \\ (bc,2) & 0 \\ (ab,1) & 1 \\ (ac+bc,3) & 1 \\ (ad+cd,4) & 0 \end{array}\right)$$

$$\xrightarrow{(c_2 + c_5 \to c_5)}$$

$$\left( \begin{array}{c|ccccc|} & (ad,2) & (bc,2) & (ab,1) & (ac{+}bc,3) & (ad{+}cd{+}bc,4) \\ \hline (1) & 1 & 0 & 0 & 0 & 0 \\ (1) & 0 & 1 & 0 & 0 & 0 \\ (0) & 0 & 0 & 1 & 1 & 1 \\ (0) & 0 & 0 & 1 & 1 & 1 \\ \hline \end{array} \; , \; \begin{array}{c|c|} & (abc,4) \\ \hline (ad,2) & 0 \\ (bc,2) & 0 \\ (ab,1) & 1 \\ (ac{+}bc,3) & 1 \\ (ad{+}cd{+}bc,4) & 0 \\ \hline \end{array} \right)$$

$$\xrightarrow{(r_3 + r_4 \to r_4)}$$

$$\left( \begin{array}{c|ccccc|} & (ad,2) & (bc,2) & (ab,1) & (ac{+}bc,3) & (ad{+}cd{+}bc,4) \\ \hline (1) & 1 & 0 & 0 & 0 & 0 \\ (1) & 0 & 1 & 0 & 0 & 0 \\ (0) & 0 & 0 & 1 & 1 & 1 \\ (0) & 0 & 0 & 0 & 0 & 0 \\ \hline \end{array} \; , \; \begin{array}{c|c|} & (abc,4) \\ \hline (ad,2) & 0 \\ (bc,2) & 0 \\ (ab,1) & 1 \\ (ac{+}bc,3) & 1 \\ (ad{+}cd{+}bc,4) & 0 \\ \hline \end{array} \right)$$

$$\xrightarrow{(c_3 + c_4 \to c_4)}$$

$$\left( \begin{array}{c|ccccc|} & (ad,2) & (bc,2) & (ab,1) & (ac{+}bc{+}ab,3) & (ad{+}cd{+}bc,4) \\ \hline (1) & 1 & 0 & 0 & 0 & 0 \\ (1) & 0 & 1 & 0 & 0 & 0 \\ (0) & 0 & 0 & 1 & 0 & 1 \\ (0) & 0 & 0 & 0 & 0 & 0 \\ \hline \end{array} \; , \; \begin{array}{c|c|} & (abc,4) \\ \hline (ad,2) & 0 \\ (bc,2) & 0 \\ (ab,1) & 0 \\ (ac{+}bc{+}ab,3) & 1 \\ (ad{+}cd{+}bc,4) & 0 \\ \hline \end{array} \right)$$

$$\xrightarrow{(c_3 + c_5 \to c_5)}$$

$$\left( \begin{array}{c|ccccc|} & (ad,2) & (bc,2) & (ab,1) & (ac{+}bc{+}ab,3) & (ad{+}cd{+}bc{+}ab,4) \\ \hline (1) & 1 & 0 & 0 & 0 & 0 \\ (1) & 0 & 1 & 0 & 0 & 0 \\ (0) & 0 & 0 & 1 & 0 & 0 \\ (0) & 0 & 0 & 0 & 0 & 0 \\ \hline \end{array} \; , \; \begin{array}{c|c|} & (abc,4) \\ \hline (ad,2) & 0 \\ (bc,2) & 0 \\ (ab,1) & 0 \\ (ac{+}bc{+}ab,3) & 1 \\ (ad{+}cd{+}bc{+}ab,4) & 0 \\ \hline \end{array} \right).$$

*Finally, we transformed our matrices into the form we were looking for, therefore, we are ready to compute* $\ker(\partial_1)$ *and* $\mathrm{Im}(\partial_2)$ *and the corresponding homology group.*

*From the left matrix we get that*

$$\partial_1(ac + bc + ab) = 0, \quad \partial_1(ad + cd + bc + ab) = 0.$$

*From these two equalities it follows that* $\ker(\partial_1)$ *is isomorphic to* $(X, \rho)$*, where* $X = \langle ac + bc + ab, ad + cd + bc + ab \rangle$*, with*

$$\rho(ac + bc + ab) = 3, \text{ and } \rho(ad + cd + bc + ab) = 4.$$

*Meanwhile, observing the persistent linear map*

$$(\partial_2)_r \colon C_2(\mathcal{F})_r \to \ker(\partial_1)_r,$$

*represented by the matrix*

$$\begin{array}{r c} & (abc,4) \\ (ac+bc+ab,3) & \left[\begin{matrix} 1 \\ 0 \end{matrix}\right. \\ (ad+cd+bc+ab,4) & \end{array}$$

$$\begin{array}{rc} & {\scriptstyle (abc,4)} \\ {\scriptstyle (ac+bc+ab,3)} \atop {\scriptstyle (ad+cd+bc+ab,4)} & \begin{bmatrix} 1 \\ 0 \end{bmatrix} \end{array},$$

*we read off that the cycle* $ad + cd + bc + ab$ *appearing at time* 4 *never becomes a boundary, since the corresponding entry in the matrix on the right is* 0*. The cycle* $ac + bc + ab$ *appearing at time* 3*, has corresponding entry equal to* 1*, therefore it becomes a boundary of the* 2*-simplex* $abc$ *at time* 4*. According to Theorem 3.33, we conclude that the* 1*-dimensional homology group is isomorphic to* $P(4, \infty) \oplus P(3, 4)$*.*

### 3.3.2 Persistent Diagrams and Barcodes

As we proved in Theorem 3.33, each finitely presented persistent vector space has a unique decomposition into interval persistent vector spaces. These are in one-to-one correspondence with finite subsets, with multiplicity, of the set

$$\{(a, b) \mid a \in [0, +\infty),\ b \in [0, +\infty] \text{ and } a < b\}.$$

One can give a visual representation of them in two distinct ways.

- **Persistent Barcodes**: Families of intervals on the non-negative real line.

  **Definition 3.40** *Let* $\{V_r\}_r = \bigoplus_{i=1}^n P(a_i, b_i)$ *be a finitely generated free persistent vector space over a field k. Then the* ***persistent barcode*** *of* $\{V_r\}_{r \in \mathbb{R}}$ *is the collection of intervals* $[a_i, b_i)$*, for* $i \in \{1, \ldots, n\}$*.*

- **Persistent Diagram**: Collection of points in the subset

  $$\{(x, y) \in \mathbb{R}^2 \mid x \geqslant 0 \text{ and } x \leqslant y\}$$

  of the first quadrant in the $(x, y)$-plane.

  **Definition 3.41** *Let* $\{V_r\}_r = \bigoplus_{i=1}^n P(a_i, b_i)$ *be a finitely generated free persistent vector space over a field k. Then the* ***persistent diagram*** *of* $\{V_r\}_{r \in \mathbb{R}}$ *is a multiset of points* $\{(a_i, b_i)\}_{1 \leqslant i \leqslant n}$ *in* $\mathbb{R}^2$ *above the diagonal.*

Persistent barcodes often consist of short and long intervals. These have different interpretations: short intervals mostly represent noise and longer intervals correspond to underlying geometric features.

**Example 3.42** *Let $\{V_r\}_r$ be a finitely generated free persistent vector space with decomposition*

$$\{V_r\}_r = P(1,5) \oplus P(2,3) \oplus P(2,4) \oplus P(3,5) \oplus P(1,2) \oplus P(0,3).$$

*In Figure 3.3 we see the persistent barcode and persistent diagram associated to the persistent vector space $\{V_r\}_r$.*

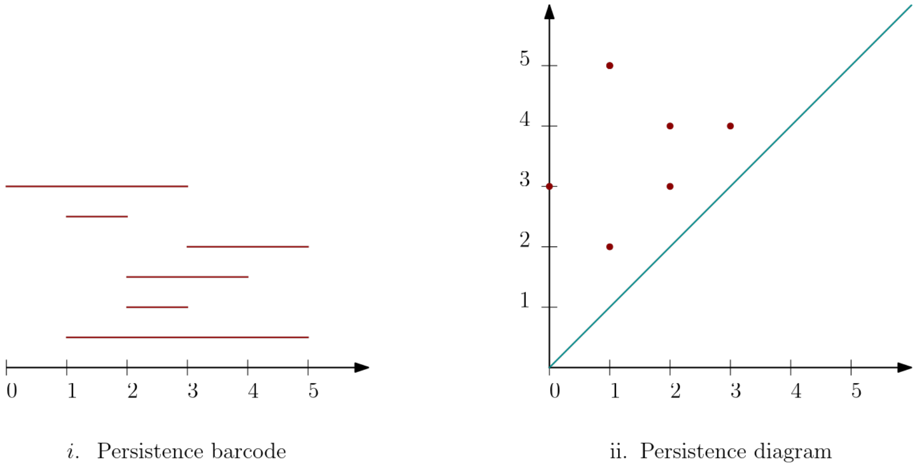


**Figure 3.3:** Barcode and diagram of the persistent vector space from Example 3.42.

## 3.4 Bottleneck Distance

After associating persistent barcodes and diagrams to finite metric spaces, we want to measure the similarity between persistence barcodes and the influence that small changes can have on a point cloud. Before the construction of such a metric, we first need to introduce the notion of distance between intervals.

**Definition 3.43** *Let $I = [x_1, y_1]$ and $J = [x_1, y_2]$ be two intervals in $\mathbb{R}$. The $l^\infty$**-distance between** $I$ **and** $J$ is given by*

$$\Delta(I, J) := \max(|x_2 - x_1|, |y_2 - y_1|).$$

*For a single interval $I = [x, y]$*

$$\lambda(I) = \frac{y - x}{2}$$

*defines the $l^\infty$-distance to the closest interval in $\{[x, x) \mid x \in \mathbb{R}\}$ to $I$.*

**Remark 3.44** *All the mentioned intervals can be regarded as points in $\mathbb{R}^2$.*

**Definition 3.45** *Let $\mathcal{I} = \{I_\alpha\}_{\alpha \in A}$ and $\mathcal{J} = \{J_\beta\}_{\beta \in B}$ be two families of intervals for finite sets A and B, and let $\varphi \colon A' \to B'$ be a bijection, where $A' \subseteq A$ and $B' \subseteq B$. The* ***penalty of*** $\varphi$ *is defined as*

$$P(\varphi) = \max\{\max_{\alpha \in A'}(\Delta(I_\alpha, J_{\varphi(\alpha)})), \max_{\alpha \in A \setminus A'}(\lambda(I_\alpha)), \max_{\beta \in B \setminus B'}(\lambda(I_\beta))\}.$$

At this point we introduce the **bottleneck distance**, which measures the distance between two families of intervals by finding the minimal distance between points and allowing unmatched points to be matched with the diagonal of $\mathbb{R}^2$.

**Definition 3.46** *Let $\mathcal{I} = \{I_\alpha\}_{\alpha \in A}$ and $\mathcal{J} = \{J_\beta\}_{\beta \in B}$ be two families of intervals for finite sets A and B, the bottleneck distance $d_\infty(\mathcal{I}, \mathcal{J})$ is defined as*

$$d_\infty(\mathcal{I}, \mathcal{J}) := \min_{\varphi} P(\varphi),$$

*where the minimum runs over all possible bijections between subsets of A and B.*

One of the most important results about the bottleneck distance is the following theorem, that guarantees stability of persistent barcodes and diagrams. We do not prove it, since it is beyond the scope of this thesis.

**Theorem 3.47** *Let X be a triangulable space and $f, g \colon X \to \mathbb{R}$ two tame functions. Then the persistent vector spaces $\{H_k(f^{-1}((-\infty, r]))\}_r$ and $\{H_k(g^{-1}((-\infty, r]))\}_r$ are finitely presented and therefore admit a barcode for every $k \in \mathbb{N}$, which we denote by $B^k_f, B^k_g$. Then*

$$d_\infty(B^k_f, B^k_g) \leqslant \|f - g\|_\infty.$$

**Example 3.48** *Consider the barcodes $\mathcal{I} = \{[1,5), [2,4)\}$ and $\mathcal{J} = \{[2,3), [4,5)\}$. We have a bijection $\rho$ between the empty subsets of both $\mathcal{I}$ and $\mathcal{J}$, which means that each point is matched to the diagonal. Next there are bijections between the singletons, where the points that are not included in the subsets are matched to the diagonal.*

$$\begin{aligned} \varphi_1 &\colon \{[1,5)\} \to \{[2,3)\}, & \psi_1 &\colon \{[1,5)\} \to \{[4,5)\}, \\ \varphi_2 &\colon \{[2,4)\} \to \{[2,3)\}, & \psi_2 &\colon \{[2,4)\} \to \{[4,5)\}. \end{aligned}$$

*For example the map $\varphi_1 \colon \{[1,5)\} \to \{[2,3)\}$ maps the point $[1,5)$ to $[2,3)$. The points that are not included in the subsets, i.e $[2,4)$ and $[4,5)$, are then matched to the diagonal. The last bijections we have are between the sets $\mathcal{I}$ and $\mathcal{J}$*

$$\begin{aligned} \omega_1 \colon \{[1,5), [2,4)\} &\to \{[2,3), [4,5)\} \\ [1,5) &\mapsto [2,3) \\ [2,4) &\mapsto [4,5), \\ \omega_2 \colon \{[1,5), [2,4)\} &\to \{[2,3), [4,5)\} \\ [1,5) &\mapsto [4,5) \\ [2,4) &\mapsto [2,3). \end{aligned}$$

*Next, we want to compute the penalty of the bijections found.*

$$\begin{aligned}P(\rho) &= \max\{\max(\lambda([1,5)),\lambda([2,4))),\max(\lambda([2,3)),\lambda([4,5)))\} = \\ &= \max\{\max(\frac{5-1}{2},\frac{4-2}{2}),\max(\frac{3-2}{2},\frac{5-4}{2})\} = \\ &= \max\{\max(2,1),\max(\frac{1}{2},\frac{1}{2})\} = 2.\end{aligned}$$

$$\begin{aligned}P(\varphi_1) &= \max\{\Delta([1,5),[2,3)),\lambda([2,4)),\lambda([4,5))\} = \\ &= \max\{\max(|2-1|,|5-3|),\frac{4-2}{2},\frac{5-4}{2}\} = \\ &= \max\{\max(1,2),1,\frac{1}{2}\} = 2.\end{aligned}$$

$$\begin{aligned}P(\psi_1) &= \max\{\Delta([1,5),[4,5)),\lambda([2,4)),\lambda([2,3))\} = \\ &= \max\{\max(|4-1|,|5-5|),\frac{4-2}{2},\frac{3-2}{2}\} = \\ &= \max\{\max(3,0),1,\frac{1}{2}\} = 3.\end{aligned}$$

$$\begin{aligned}P(\varphi_2) &= \max\{\Delta([2,4),[2,3)),\lambda([1,5)),\lambda([4,5))\} = \\ &= \max\{\max(|2-2|,|4-3|),\frac{5-1}{2},\frac{5-4}{2}\} = \\ &= \max\{\max(0,1),2,\frac{1}{2}\} = 2.\end{aligned}$$

$$\begin{aligned}P(\psi_2) &= \max\{\Delta([2,4),[4,5)),\lambda([1,5)),\lambda([2,3))\} = \\ &= \max\{\max(|4-2|,|5-4|),\frac{5-1}{2},\frac{3-2}{2}\} = \\ &= \max\{\max(2,1),2,\frac{1}{2}\} = 2.\end{aligned}$$

$$\begin{aligned}P(\omega_1) &= \max\{\Delta([1,5),[2,3)),\Delta([2,4),[4,5))\} = \\ &= \max\{\max(|2-1|,|5-3|),\max(|4-2|,|5-4|)\} = \\ &= \max\{\max(1,2),\max(2,1)\} = \\ &= \max\{2,2\} = 2.\end{aligned}$$

$$\begin{aligned}P(\omega_2) &= \max\{\Delta([1,5),[4,5)),\Delta([2,4),[2,3))\} = \\ &= \max\{\max(|4-1|,|5-5|),\max(|2-2|,|4-3|)\} = \\ &= \max\{\max(3,0),\max(0,1)\} = \\ &= \max\{3,1\} = 3.\end{aligned}$$

*Now that we are done with the penalties of the functions, we are ready to calculate the bottleneck distance between the two families $\mathcal{I}$ and $\mathcal{J}$.*

$$d_\infty(\mathcal{I}, \mathcal{J}) = \min\{P(\rho), P(\varphi_1), P(\psi_1), P(\varphi_2), P(\psi_2), P(\omega_1), P(\omega_2)\} = 2$$

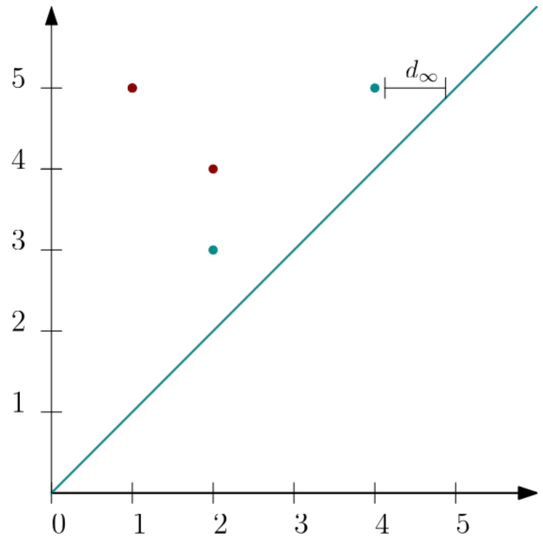


**Figure 3.4:** Persistent diagram of the two families of intervals discussed in Example 3.48.

## Chapter 4

# Neuronal Morphology Characterisation

A key aspect of understanding the structure and functionality of the brain lies in the comprehension of neurons and their morphology. With the help of topological data analysis, we introduce the **Topological Morphology Descriptor** (TMD), an algorithm that assigns a barcode to a merge tree. This tool has proven useful for the classification of neuronal morphologies since these barcodes allow for an accurate comparison of neurons. We primarily focus on applying the TMD on **pyramidal cells**, which are a type of neuron associated with advanced cognitive functions.

In addition to the TMD, we introduce an algorithm that approximates the right inverse of the TMD called **Topological Neuron Synthesis** (TNS). The purpose of the TNS is to recreate the neuronal tree from the barcode created by the TMD. This chapter is mostly based on the papers *From Trees to Barcodes and Back Again: theoretical and statistical perspectives* and *From Trees to Barcodes and Back Again II: Combinatorial and Probabilistic Aspects of a Topological Inverse Problem* [17, 6]. To conclude, we review some results that appear in *A Topological Representation of Branching Neuronal Morphologies* [16] with the TMD algorithm for the classification of neuronal morphologies.

## 4.1 Neurons

The brain is one of the most complex parts of our body and together with the nervous system are composed of different types of cells. The fundamental units are called **neurons**. Neurons are responsible for our movements, ideas, sensations, and memories, which arise from the electrical signals that get passed between them. This electrical event is generated in the *axon* and is called *action potential*. It signals that the neuron is active.
A neuron is composed of three parts: the soma, the dendrites, and the axon.

- The *soma* is the cell body of the neuron. Here lies the nucleus and the

DNA of the neuron. Additionally, all the proteins transported through the axon and dendrites are produced in the soma.

- The *dendrites* are the receiving part of the neuron. They receive synaptic inputs from the axon and the sum of dendritic inputs determines if there will be an action potential.
- The *axon* is a long thin structure where the action potential is generated and is also the transmitting part of a neuron. The action potential travels down the axon to cause the release of neurotransmitters into the synapse.

This allows the neuron to communicate with other neurons. In this thesis we concentrate on neuronal morphology, i.e. is the shape and structure of the neuron [25, 20].

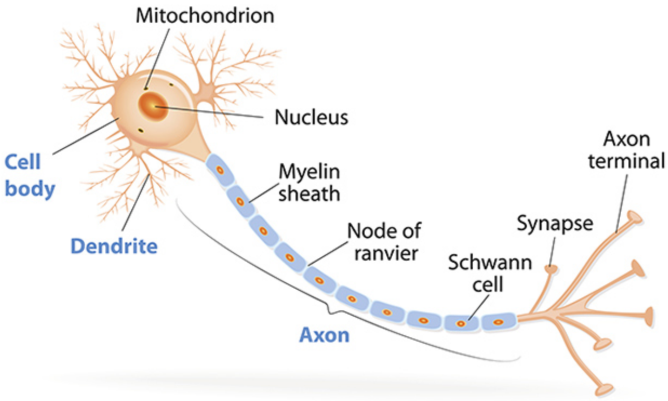


**Figure 4.1:** Representation of a neuron with soma (cell body), dendrites, and axon. [24].

In particular, the neurons we are mostly interested in are called **pyramidal cells**. This is a particular type of neuron associated with advanced cognitive functions found in the cerebral cortex of most mammalian brains. Pyramidal cells belong to the family of excitatory neurons realizing the neurotransmitter glutamate. They are characterized by their distinct apical dendritic tree, longer dendrites emerging from the point end of the soma, and basal dendritic tree, shorter dendrite coming from its rounded base, and the pyramidal shape of their soma, from which their name comes. They cover two-thirds of all neurons present in the mammalian cerebral cortex. Pyramidal cells all appear very similar to each other, however, they come up in different shapes and sometimes also happen to function differently. As one might already recognize from Figure 4.2 all the illustrated pyramidal cells have the same structure but at the same time, they are all different [22, 1].

This is where we apply the algorithm introduced in Section 4.2.3, to analyze how the pyramidal cells differ from each other and to properly classify them.

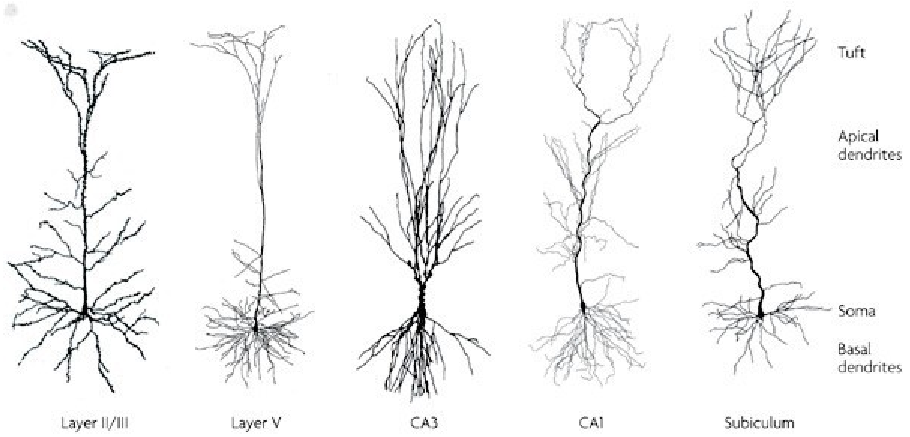


**Figure 4.2:** Pyramidal cells taken from different cortical areas each presenting the apical tuft, apical and basal dendrites and soma [22].

## 4.2 From Trees to Barcodes

In this section, we use persistent homology to characterize the topological and geometric features of a merge tree $(T, h)$. To this purpose, we use the sublevel set filtration of the height function $h$ of the considered merge tree embedded in $\mathbb{R}^3$. Most importantly, we introduce the TMD algorithm, which can be applied not only to merge trees but also to geometric trees, providing a method for their categorization.

For the application of the TMD algorithm to neurons, we first need to create a corresponding digital reconstruction. Such a digital reconstruction is done by sampling a set of points in $\mathbb{R}^3$ along each branch, together with edges connecting adjacent points. This results in a combinatorial merge tree preserving the same morphological information as the neuronal cells.

### 4.2.1 Elder Rule

The goal of this section is to explain how to create a barcode from a merge tree. We provide a way of constructing barcodes from merge trees via the decomposition of branches.

**Definition 4.1 (Elder Rule)** *Let $(T, h)$ be a merge tree, each leaf node marks the first coordinate of a pair with $h(v)$. If two leaf nodes $v_i, v_j$ with $h(v_i) > h(v_j)$ share an ancestor $v_k$, then the branch with the bigger h-value dies, since it is the one born last, creating a bar $[h(v_i), h(v_k))$ in the barcode.*

From this construction, we recognize that each leaf node marks the beginning of a bar in the barcode and such always ends at an internal node.

**Example 4.2** *The merge tree* $(T, h)$ *in Figure 4.3 is composed by the vertex set* $V(T) = \{b_0, b_1, b_2, b_3, d_0, d_1, d_2, d_3\}$, *where* $b_i$ *denotes the leaf of the tree and* $d_i$ *the inner vertices for* $0 \leqslant i \leqslant 3$. *The application of the elder rule to* $(T, h)$ *creates the barcode* $B = \{[b_i, d_i)\}_{0 \leqslant i \leqslant 3}$ *illustrated in the figure.*

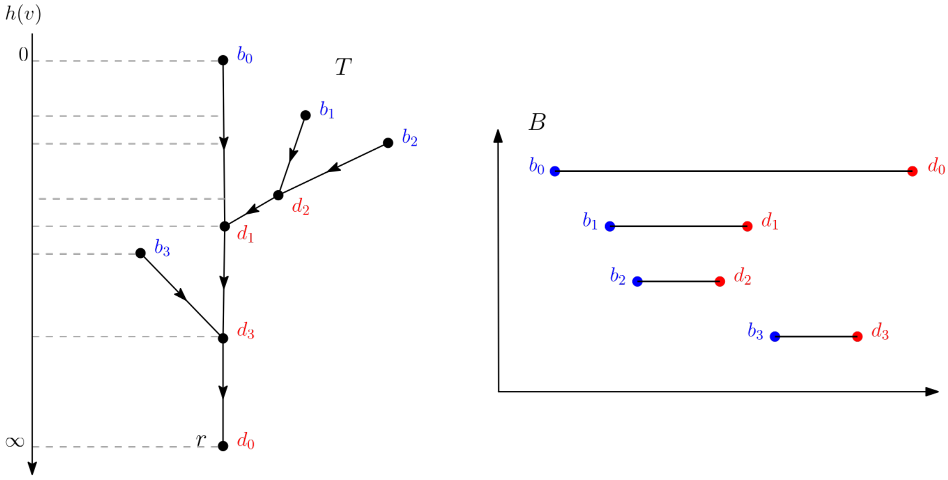


**Figure 4.3:** Example of a Barcode $B$ generated after the application of the Elder Rule on the merge tree $(T, h)$.

The Elder Rule can also be defined for combinatorial merge trees, called the **Combinatorial Elder Rule**. This replaces the function $h$ with $L_i$ and $L_l$ and instead of giving back a barcode, it returns a pair of labels $(L_l(v_i), L_i(v_j))$ corresponding to a combinatorial permutation assigning to each birth label a death label.

**Definition 4.3 (Combinatorial Elder Rule)** *Let* $(T, L_l, L_i)$ *be a combinatorial merge tree, where each leaf node* $v$ *marks the first coordinate of a pair label* $L_l(v)$. *If two leaf nodes* $v_i, v_j$ *with* $L_l(v_i) < L_l(v_j)$ *share an ancestor* $v_k$, *then* $v_k$ *gets paired with* $v_i$, *since it has the smallest label, creating the pair* $(L_l(v_i), L_i(v_k))$. *From this rule, the leaf with the smallest label gets paired with the root, constructing* $(\min_i L_l(v_i), \infty)$.

### 4.2.2 Strict Barcodes

In our application, we produce a particular type of barcodes called *strict barcodes*.

**Definition 4.4** *A **strict barcode** is a barcode* $B = \{[b_i, d_i)\}_{i \in \{0,\ldots,n\}}$, *such that*

- *the bar $[b_0, d_0)$ contains all the others:*

$$b_0 < b_i \text{ and } d_0 > d_i \quad \forall i \in \{1, \ldots, n\},$$

- *No bars are born or die at the same time:*

$$b_i \neq b_j \text{ and } d_i \neq d_j \text{ if } i \neq j.$$

**Remark 4.5** *We refer to $b_i$ as the **birth** of a feature and to $d_i$ as its **death**.*

**Example 4.6** *The barcode shown in Figure 3.3 is not a strict barcode. The bars $[2,3)$, $[2,4)$ and $[1,2)$, $[1,5)$ are born at the same time and the bars $[3,5)$, $[1,5)$ die at the same time.*

**Remark 4.7** *The barcode of a merge tree is always strict.*

We denote the set of all barcodes by $\mathcal{B}$ and the set of all *strict* barcodes by $\mathcal{B}^{st}$. With $\mathcal{B}_n \subseteq \mathcal{B}$ we mean all the barcodes with $n+1$ bars and with $\mathcal{B}_n^{st} \subseteq \mathcal{B}^{st}$ we mean all the *strict* barcodes with $n+1$ bars.

**Remark 4.8** *In a strict barcode, the birth times admit a total order, hence w.l.o.g we always assume that $b_0 < b_1 < \ldots < b_n$.*

**Remark 4.9** *We use the convention that for a bar $[b_i, d_i) \subseteq \mathbb{R}$ with $i \geqslant 0$, the corresponding point in $\mathbb{R}^2$ are always given by $(d_i, b_i) \in \mathbb{R}^2$, implying that all the points have to lie below the diagonal.*

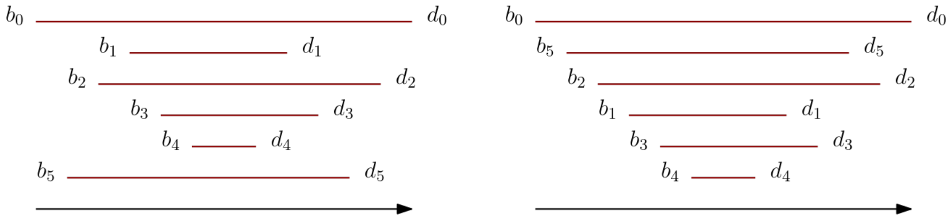


**Figure 4.4:** Example of a strict barcode on the left and the same barcode ordered by birthtime on the right.

**Definition 4.10** *Two strict barcodes $B = \{[b_i, d_i)\}_{0 \leqslant i \leqslant n}$ and $B' = \{[b_i', d_i')\}_{0 \leqslant i \leqslant n}$ in $\mathcal{B}_n^{st}$ are **equivalent**, if their death occurrs in the same order i.e. if for all $1 \leqslant i \neq j \leqslant n$*

$$B \underset{bar}{\sim} B' :\iff (d_i < d_j \iff d_i' < d_j').$$

*This relation defines an equivalence relation on $\mathcal{B}_n^{st}$.*

**Remark 4.11** *Because the birth ordering of a strict barcode delivers an ordering of the deaths, there is a bijection from the set of equivalence classes of strict barcodes*

*$\mathcal{B}_n^{st}$ with $n+1$ bars to the symmetric group $\mathcal{S}_n$ given by*

$$\begin{aligned} {\mathcal{B}_n^{st}}/{\sim} &\longrightarrow \mathcal{S}_n, \\ B &\longmapsto \rho_B \colon \{1,\dots,n\} \to \{1,\dots,n\} \\ &\qquad\qquad\qquad i \mapsto |\{j \in \{1,\dots,n\} \mid d_i < d_j\}|. \end{aligned}$$

*We denote the equivalence class containing a strict barcode $B = \{(b_i, d_i)\}_{0 \leqslant i \leqslant n}$ by $(i_1, \dots, i_n)$, where $d_{i_k} > d_{i_{k+1}}$ for all $1 \leqslant k \leqslant n$. The permutation $\rho_b$ is often called* ***associated permutation of*** *B or* ***combinatorial barcode****.*

**Example 4.12** *For the strict barcode $B = \{[b_i, d_i)\}_{0 \leqslant i \leqslant 5}$ in Figure 4.5 we can find a permutation in $\mathcal{S}_5$ determining the equivalence class that B belongs to. In this case $\rho_B = (21435)$.*

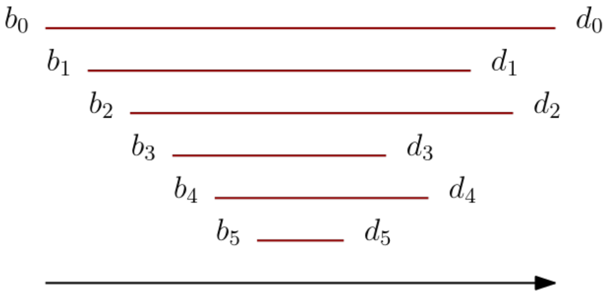


**Figure 4.5:** Strict barcode $B$ with corresponding equivalence class $\rho_B = (21435)$.

**Definition 4.13** *Let $B = \{[b_i, d_i)\}_{0 \leqslant i \leqslant n}, B' = \{[b_i', d_i')\}_{0 \leqslant i \leqslant n} \in \mathcal{B}_n^{st}$ be strict barcodes with $n+1$ bars. Then B and $B'$ are said to be* ***combinatorially equivalent*** *if they have the same associated permutations $\rho_B$ and $\rho_{B'}$.*

Applying the Elder Rule introduced in Definition 4.3 to a merge tree associates a permutation to the constructed barcode.

**Proposition 4.14** *If $T$ and $T'$ are combinatorially equivalent merge trees, then their corresponding barcodes $B$ and $B'$ are combinatorially equivalent.*

**Proof** Since $T$ and $T'$ are combinatorial equivalent merge trees we know there is a graph isomorphism $\varphi \colon T \to T'$ preserving the order of birth and death labels. Moreover, since $\varphi$ is a graph isomorphism, the unique sequence of edges connecting a pair of nodes in $T$ must be sent to the same sequence of edges connecting those nodes in $T'$, the adjacency relation is therefore preserved, which implies that if the Elder Rule pairs the $i$-th birth with the $j$-th death in $T$, then so does it in $T'$. ☐

At this point, we want to endow the set of all strict barcodes with a metric. To this purpose, we adapt the bottleneck distance to strict barcodes introduced in Chapter 3.4.

**Definition 4.15** *Given two strict barcodes $B, B' \in \mathcal{B}_n^{st}$ we define the* ***modified bottleneck distance*** *as*

$$\tilde{d}_\infty(B, B') = \inf_{\gamma \in \mathcal{S}^n} \sup_{0 \leqslant i \leqslant n} \left( |b_i - b'_{\gamma(i)}| + |d_i - d'_{\gamma(i)}| \right).$$

**Remark 4.16** *One might notice that in this adapted definition of bottleneck distance, there is no connection to the diagonal $\{[x, x) \in \mathbb{R}^2 \mid x \in \mathbb{R}\}$, therefore we cannot match unmatched points to the diagonal. Moreover, this new definition of bottleneck distance is only well-defined when both barcodes have the same number of bars. However, this is not an issue, since we only compare barcodes $B \in \mathcal{B}_n^{st}$ with their transformation under $TMD \circ TNS(B)$, which always retain the same number of bars.*

### 4.2.3 Topological Morphology Descriptor

The **Topological Morphology Descriptor**(TMD) is the surjective function

$$\text{TMD}\colon \mathcal{T} \to \mathcal{B}.$$

It captures topological and geometric features from geometric trees and saves them into a barcode, where each bar represents a branch of the tree. This is mostly used for the characterization of neuronal morphologies by replacing geometric trees with the digital representation of neurons. The TMD applies a similar procedure as the Combinatorial Elder Rule described in Definition 4.3 on geometric trees. It is recursively defined as follows.

Let $T$ be a finite rooted tree with root $r$, set $N$ of vertices, and $L \subseteq N$ the subset of leaves. Moreover, let

$$\delta\colon N \to \mathbb{R}^+, v \mapsto \|v - r\|_2$$

be the function assigning to a vertex its Euclidean distance to the root. Note that $\delta$ fulfills the definition of a height function.

Let $\mu\colon N \to \mathbb{R}$ be the function defined by

$$\mu(v) = \begin{cases} \max\{\delta(l) \mid l \in L_v\}, & v \in N \setminus L, \\ \delta(v), & v \in L, \end{cases}$$

where $L_v$ denotes the set of leaves of the subtree $T$ with root at $v$. The function $\mu$ creates an ordering of the children of any vertex of $T$. For $v_1, v_2 \in N$ we say that $v_1$ is *younger* than $v_2$, if $\mu(v_1) < \mu(v_2)$.

**Example 4.17** *The tree in Figure 4.6 has set of leaves L and set of inner vertices I, where $N = I \cup L$:*

$$L = \{b_0, b_1, b_2, b_3\}, \quad I = \{d_0, d_1, d_2, d_3\}. \tag{4.1}$$

*Take $b_0 \in L$, then $\mu(b_0) = \delta(b_0)$ corresponds to the Euclidean distance to the root. On the other hand, if we observe $d_1 \in I$, we see*

$$\mu(d_1) = \max\{\delta(l) \mid l \in L_{d_1}\} = \max\{\delta(b_1), \delta(b_2)\} = \delta(b_2),$$

*since $b_2$ is further away from the root than $b_1$.*

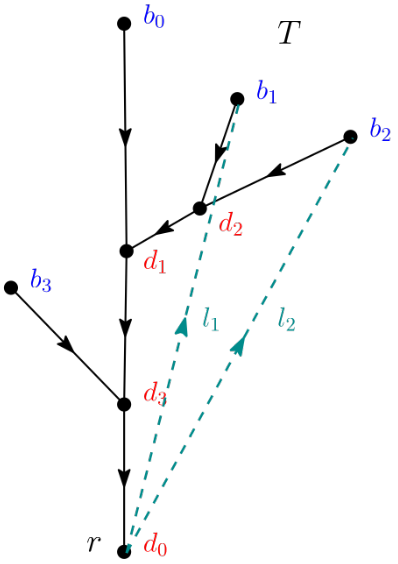


**Figure 4.6:** Tree with Euclidean distance $l_1, l_2$ from the vertices $b_1, b_2$ to the root $r$ discussed in Example 4.17.

The TMD algorithm works as follows:

Let $A$ be a set of vertices, called the **active vertices** and $B$ a barcode. They are initially set to $A = L$, $B = \emptyset$.

- Take a leaf $l \in L$ and walk along the unique path to the root $r$.
- When encountering a branching point $b$, apply the Elder Rule.
- Remove from $A$ all the children of $b$ and add $b$ to $A$.
- Add one bar to the barcode $B$ for each child of $b$ removed from $A$, except the longest bar.
- Apply this procedure iteratively to each vertex, until $A = \{r, l\}$, where $\mu(l) = \max_{l' \in L} \mu(l')$.

Each child removed from $A$ corresponds to a path from some leaf $l$ to a branching point $b$, recorded as a bar $[\delta(b), \delta(l))$. For the last remaining leaf $l$ its $\mu$-value will be maximal, resulting in the bar $[\delta(r), \delta(l)) = [0, \delta(l))$.

For a digitally reconstructed neuron $T$ and a function $\delta$, denoting the distance from the soma, TMD$(T)$ is a strict barcode, since the probability that two

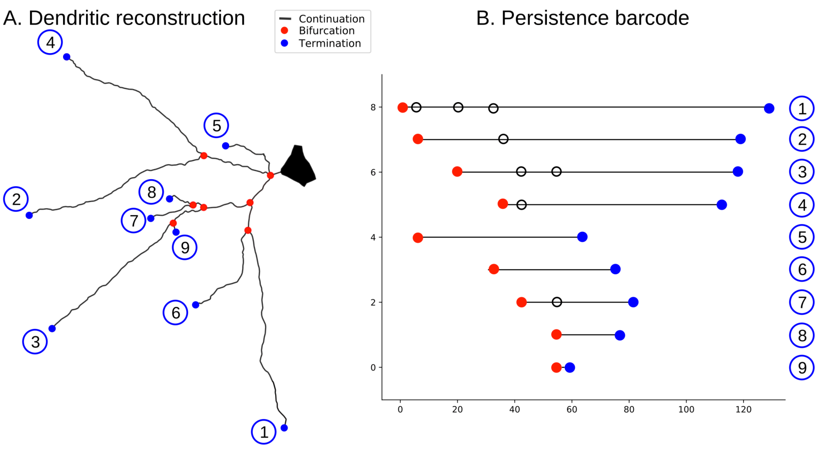


**Figure 4.7:** Computation of the TMD algorithm on the digital reconstruction of a neuron results in the persistent barcode $B$. The red disks represent birth and the blue ones' death points. The circles on the bar represent at which branch the bifurcation takes place [17].

branch points or two leaves have the same distance from the soma is almost 0.

**Definition 4.18** *Two geometric trees $T, T' \in \mathcal{T}$ are called* ***TMD-equivalent*** *if their generated barcodes are equivalent, i.e.*

$$T \underset{tmd}{\sim} T' :\iff TMD(T) \underset{bar}{\sim} TMD(T').$$

**Remark 4.19** *The output of the TMD algorithm is the 0-dimensional barcode, of the distance function $\delta$. For a barcode $B = \{[b_i, d_i)\}_{1 \leqslant i \leqslant n}$ each bar $[b_i, d_i)$ corresponds to a connected component in the sublevel sets $\delta^{-1}([0,t))$, which is equal to a branch of the tree.*

## 4.3 Topological Neuronal Synthesis

In the last section, we introduced the TMD algorithm, which assigns a strict barcode to a geometric tree. We now want to stochastically reconstruct the geometric tree $T$ we began with, such that the input barcode $B$ is then closely related to the generated barcode $TMD(T)$. This process is called **Topological Neuronal Synthesis** (TNS), which is mostly used for the digital reconstruction of brain circuitry by stochastically generating synthetic neurons.

For an input barcode $B$, the TNS algorithm goes through three steps while generating a geometric tree:

1. Initiation of growth,
2. Elongation,

3. Branching / Termination.

First, it initiates the growth of the tree, then proceeds to elongate it as a directed random walk. At each step, a growing tip is assigned probabilities to terminate, bifurcate or elongate depending on the chosen bar and the distance from the tip to the root. As soon as the bar is used, it is then removed from $B$. This process is repeated until the barcode $B$ is empty.

We now give further explanation of the steps of branching/termination and elongation.

### 4.3.1 Bifurcation and Termination Process

The branching process in the TNS algorithm is based on the concept of a Galton-Watson tree, which is a recursively generated finite rooted tree. To generate such a tree, we independently sample the number of offspring at each step, from a distribution, and since geometric trees only consist of elongations, bifurcations, and terminations the only accepted values are:

- zero $\leftrightarrow$ termination,
- one $\leftrightarrow$ continuation,
- two $\leftrightarrow$ bifurcation.

The probability of bifurcating/terminating depends on the distance of the growing tip from the root, which would transform the tree from a combinatorial tree to a geometric tree. The probabilities to bifurcate and terminate are sampled from an exponential distribution $e^{-\lambda x}$, with free parameter $\lambda$.

For a barcode $B$, the bifurcation/termination step of the growth process of a geometric tree associated to $B$ works as follows:

- Assign a bar $[b_i, d_i)$ taken from the barcode $B$ to each growing tip and a bifurcation angle $a_i$, encoded in the barcode[1].
- Check if a branch terminates or bifurcates by randomly sampling a number $r$ in the uniform distribution $U(0,1)$ and compare it to

  $$P_B(\text{bifurcation} \mid d_{tip}) = e^{\lambda(d_{tip}-b_i)},$$

  or

  $$P_T(\text{termination} \mid d_{tip}) = e^{\lambda(d_{tip}-d_i)},$$

  where in both cases $d_{tip}$ is the distance from the growing tip to the root. If

  $$r \leqslant P_B(\text{bifurcation} \mid d_{tip}) = e^{\lambda(d_{tip-b_i})},$$

---

[1] We apply the TNS exclusively to barcodes that have been generatedby the application of the TMD on a geometric tree. Therefore, the starting point $b_i$, the death point $d_i$ and the bifurcation angle $a_i$ on a bar are measured and encoded while the TMD runs through the geometric tree.

or
$$r \leqslant P_T(\text{termination} \mid d_{tip}) = e^{\lambda(d_{tip}-d_i)},$$

then respectively either a bifurcation or a termination occurs. In case none of the two cases holds, the growing tip continues its elongation.

- As soon as a bar is used, it is removed from the barcode to prevent re-sampling of the same conditional probability.

If $d_{tip} < b_i$, then $P_B(\text{bifurcation} \mid d_{tip}) < 1$ and increases as $b_i$ is approached. The branch will surely bifurcate as soon as $d_{tip} = b_i$, since in that case $P_B(\text{bifurcation} \mid d_{tip}) = 1$. Moreover, whenever a bifurcation takes place, then the directions of the new branches depend on the bifurcation angle $a_i$. Similarly, if $d_{tip} < d_i$, then $P_T(\text{termination} \mid d_{tip}) < 1$, hence the growing tip will terminate as soon as $d_{tip} = d_i$ and therefore $P_T(\text{termination} \mid d_{tip}) = 1$.

These last two cases we analyzed strongly depend on the choice of $\lambda$ since it controls the slope of the probability distribution for bifurcation and termination. Its choice has thus to be done carefully. If we choose $\lambda$ to be very high, then the resulting geometric tree would be identical to the input, but if it is chosen to be very low, then the resulting geometric tree is completely random and independent of the input. Assuming that the growing step size is $L$, we choose $\lambda \approx L$, which ensures biologically appropriate variance. The step size $L$ is mostly chosen to be 1, since then the bifurcation and termination points, although stochastically chosen, are strongly correlated with the input barcode.

### 4.3.2 Elongation

To embed a synthesized tree into $\mathbb{R}^3$, we need to assign to each of its **segment**, edge between two consecutive vertices, a direction, called the **direction of a segment**. This is denoted as a vector $\vec{d}$ and is the weighted sum of three unit vectors:

1. The **cumulative memory** $\vec{m}$,
2. A **target vector** $\vec{t}$,
3. A **random vector** $\vec{r}$.

The cumulative memory is a weighted sum of the directions of the branch with the weight decreasing with the distance from the tip. Observe now the $k$-th segment of the growing branch, then the respective memory vector can be given by a weighted sum of the previous five segment

$$\vec{m} = \sum_{i=1}^{5} e^{1-i} v_{k-i}.$$

However, the precise choice of this function is not important as long as it decreases faster than linearly with the growing distance from the tip. The target vector $\vec{t}$ is defined at the beginning of each branch and the random vector $\vec{r}$ is sampled uniformly from $\mathbb{R}^3$.
The direction of a segment is therefore given by

$$\vec{d} = \rho\vec{r} + \tau\vec{t} + \mu\vec{m},$$

where $\rho, \tau, \mu \in \mathbb{R}$ are weight parameters with $\rho + \tau + \mu = 1$. Different combinations of the parameters $\rho, \tau, \mu$ give the possibility of creating a wide range of geometric trees.

The TNS works as a right inverse to the TMD only if the branch corresponding to a bar $[b_i, d_i)$ is attached to branches corresponding to bars $[b_j, d_j)$ such that $d_i < d_j$ and $b_i > b_j$. This restriction ensures that the Elder Rule still holds while applying the TMD transformation. This procedure then recreates a tree almost TMD-equivalent to the original.

**Example 4.20** *Figure 4.8 A gives a visual representation of the different processes that the TNS goes through for the synthesis of a neuron. Figure 4.8 B shows a detailed construction of a neuron from a given barcode. We begin constructing segments from a barcode as provided by the direction vector* $\vec{d}$ *and the probability for bifurcation, termination, or elongation depends on the distance from the root/soma. The probability of bifurcating or terminating increases as soon as one reaches the birth or the death point of a bar until this reaches one. As soon as a death point is encountered, the bar it belongs to is removed from the barcode. This process is continued until there is no bar left in the barcode.*

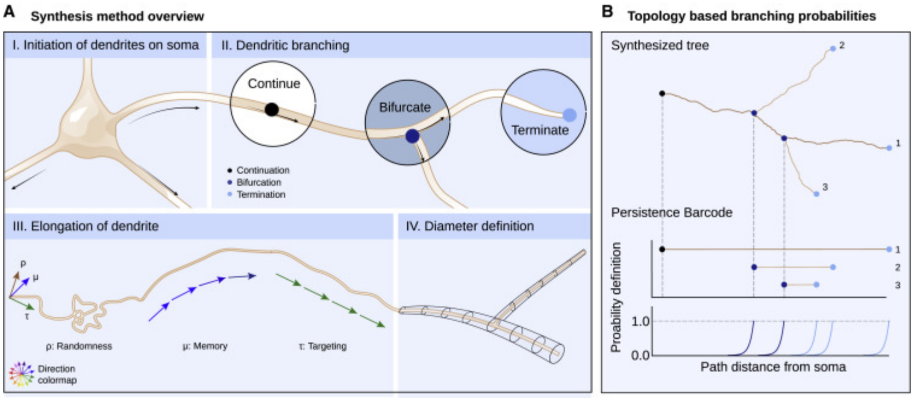


**Figure 4.8:** In (A) the illustration of the TNS process, showing: (I) Growth initiation of Soma, (II) Branching/Termination/Continuation of Dendrite, (III) Elongation of Dendrite and (IV) Diameter Definition. Part (B) represents how a tree is synthesized with the use of probability and the barcode created from the TMD algorithm. [15]

## 4.4 Tree Realization Number

In the description of the TMD, we noticed that it defines a surjective function. However, it is not injective, since a specific barcode can be generated by different trees. In this section, we want to analyze to which extent the failure of injectivity is meaningful for the classification of barcodes.

**Definition 4.21** *Let $B$ be a barcode. A geometric tree $T$ is called a* ***tree realization*** *if $TMD(T) = B$, which is equivalent to $T \in TMD^{-1}(B)$.*

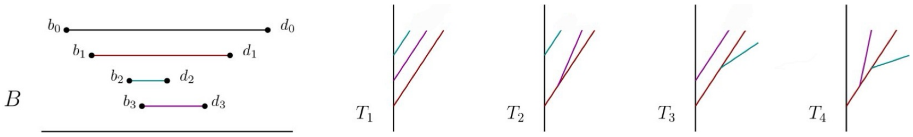


**Figure 4.9:** The trees $T_1, T_2, T_3$ are all tree realizations of the barcode $B$.

There can be a lot of trees generated from one barcode since TMD is not an injective function, this is why for any strict barcode $B$ we introduce the **set of combinatorial equivalence classes** $\mathcal{T}(B)$

$$\mathcal{T}(B) = {}^{TMD^{-1}(B)}\Big/_{\underset{comb}{\sim}} .$$

This enables us to characterize the equivalence relation on strict barcodes through $\mathcal{T}(B)$.

**Proposition 4.22** *If $B, B' \in \mathcal{B}_n^{st}$ are two strict barcodes, then*

$$B \underset{bar}{\sim} B' \iff \mathcal{T}(B) = \mathcal{T}(B').$$

**Definition 4.23** *Let $B = \{[b_i, d_i)\}_{\{0 \leqslant i \leqslant n\}} \in \mathcal{B}_n^{st}$ be a strict barcode. The* ***tree-realization number*** *of $B$ is defined as*

$$TRN(B) = |\mathcal{T}(B)|,$$

*which is the number of combinatorial equivalence classes of tree-realizations of $B$.*

**Example 4.24** *In Figure 4.9 we see that $T_1, T_2, T_3$ and $T_4$ are all the possible tree realizations of the barcode $B$, therefore*

$$TRN(B) = |\mathcal{T}(B)| = 4.$$

Although this definition is straightforward, there exists a different characterization of the tree-realization number of a barcode, using the index of a barcode. Remember that when working with strict barcodes we always assume w.l.o.g that the bars are ordered by birth.

**Definition 4.25** *Let* $B = \{[b_i, d_i)\} \in \mathcal{B}_n^{st}$ *be a strict barcode. Then for a bar* $[b_i, d_i) \in B$ *the index of the bar is given by*

$$index_i(B) = |\{j \mid b_j < b_i < d_i < d_j\}| = |\{j < i \mid < d_i < d_j\}|.$$

Intuitively the index of a bar $[b_i, d_i)$ is the number of bars that strictly contain $[b_i, d_i)$.

**Proposition 4.26** *The tree-realization number of a strict barcode* $B = \{[b_j, d_j)\}_{0 \leqslant j \leqslant n}$ *is equal to the product of the indices of its bars,*

$$TRN(B) = \prod_{1 \leqslant i \leqslant n} index_i(B).$$

**Proof** By the Elder Rule in the TMD a branch can be attached to another only if its corresponding bar is included in the other. This observation enables us to prove this proposition by using recursion on the number of bars.
We provide a brief sketch. Set $T_0 = [b_0, \infty)$, which is the trunk of the tree corresponding to the longest bar of the barcode. Since the tree is connected, we can recursively attach bars by death time, first to $T_0$, and then in the $n$-th step, we attach a branch to $T_n$ to get $T_{n+1}$, according to the Elder Rule. □

**Example 4.27** *For the computation of the tree realization number of the barcode in Figure 4.10, we first provide the index of each bar.*

$$index_1(B) = 1, \qquad index_2(B) = 2,$$
$$index_3(B) = 3, \qquad index_4(B) = 1.$$

*We can make now use of Proposition 4.26 and get that* $TRN(B)$ *is given by*

$$TRN(B) = \prod_{1 \leqslant i \leqslant 4} index_i(B) = 1 \cdot 2 \cdot 3 \cdot 1 = 6.$$

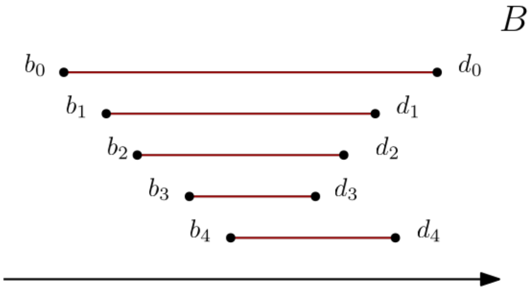


**Figure 4.10:** Strict barcode $B = \{[b_i, d_i)\}_{0 \leqslant i \leqslant 4}$ used for the computation of the TRN in Example 4.27.

**Remark 4.28** *Notice that for a strict barcode $B \in \mathcal{B}_n^{st}$ the maximal achievable tree-realization number is $n!$, which happens whenever $d_n < \ldots < d_1 < d_0$. Such a barcode is called a* ***strictly ordered*** *or a* ***Russian doll*** *barcode.*

Note that in general, the tree-realization number is not an invariant of the barcode equivalence relation, since for two barcodes $B, B' \in \mathcal{B}_n^{st}$ it **does not hold** that

$$TRN(B) = TRN(B') \Rightarrow B \sim B'.$$

However, the inverse implication holds:

$$TRN(B) \neq TRN(B') \Rightarrow B \nsim B',$$

enabling the identification of non-equivalent barcodes.

### 4.4.1 Alteration of Barcodes

At this point, we want to investigate up to which point we can interpret the TRN as an invariant and thus analyze the insertion of a bar into a barcode and the transposition of two bars in a barcode.

**Proposition 4.29 (Addition of a bar)** *Let $B = \{[b_i, d_i)\}_{0 \leqslant i \leqslant n} \in \mathcal{B}_n^{st}$ be a strict barcode and let $B' = B \cup \{[b_{n+1}, d_{n+1}\} \in \mathcal{B}_n^{st}$, where $b_{n+1} > b_i$ for all $1 \leqslant i \leqslant n$. If $d_{i_1} > \ldots > d_{i_{k-1}} > d_{n+1} > d_{i_k} > \ldots > d_{i_1 n}$, then*

$$TRN(B') = k \cdot TRN(B'). \tag{4.2}$$

**Proof** From the ordering condition imposed on $d_{n+1}$, we recognize that $[b_{n+1}, d_{n+1})$ is strictly included in exactly $k$ other bars, so its index is $k$. The result in Equation 4.2 follows from Proposition 4.26. ☐

**Example 4.30** *In Example 4.26 we saw that $TRN(B) = 6$. As one can see in Figure 4.11, we added a bar $[b_5, d_5)$ in the barcode $B$, creating a new barcode $B' = \{[b_i, d_i)\}_{0 \leqslant i \leqslant 5}$. From the picture, one can recognize that $index_5(B) = 3$, therefore from Proposition 4.26 we get that*

$$TRN(B') = \prod_{1 \leqslant i \leqslant 5} index_i(B') = 1 \cdot 2 \cdot 3 \cdot 1 \cdot 3 = 18 = 3 \cdot TRN(B),$$

*which is the result we expected using Proposition 4.29.*

**Proposition 4.31 (Permutation of deaths)** *Let $B = \{[b_i, d_i)\}_{\{0 \leqslant i \leqslant n\}} \in \mathcal{B}_n^{st}$ be a strict barcode in the equivalence class $(i_1 \cdots i_n)$. Let $B' = \{[b'_i, d'_i)\}_{\{0 \leqslant i \leqslant n\}}$ be a new barcode, such that $b_i = b'_i$ for all $i \in \{0, \ldots, n\}$ and $d_i = d'_i$ for all $i \neq i_k, i_{k+1}$, while $d_{i_k} = d'_{i_{k+1}}$ and $d_{i_{k+1}} = d'_{i_k}$, i.e. permute the deaths $d_{i_k}$ with $d_{i_{k+1}}$.*

1. *If $i_k < i_{k+1}$, then $index_{i_{k+1}}(B') = index_{i_{k+1}}(B) - 1$ and*

$$TRN(B') = \frac{TRN(B)(index_{i_{k+1}}(B) - 1)}{index_{i_{k+1}}(B)}.$$

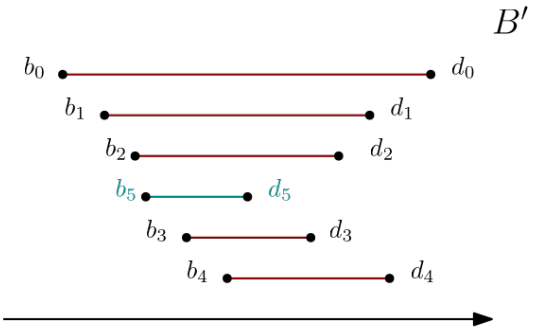


**Figure 4.11:** Alteration of the barcode $B$ given in Figure 4.10 with the addition of the bar $[b_5, d_5)$ used in Example 4.30.

2. *If* $i_k > i_{k+1}$*, then* $index_{i_{k+1}}(B') = index_{i_{k+1}}(B) + 1$ *and*

$$TRN(B') = \frac{TRN(B)(index_{i_{k+1}}(B) + 1)}{index_{i_{k+1}}(B)}.$$

**Proof** It is enough to prove the first statement since the second one follows by switching the role of $B$ and $B'$.

Assume that $i_k < i_{k+1}$, then by assumption on the barcodes $b_{i_k} < b_{i_{k+1}}$. Since $B$ is in the equivalence class of $(i_1 \cdots i_n)$, then $d_{i_{k+1}} < d_{i_k}$, which implies that $[b_{i_{k+1}}, d_{i_{k+1}}) \subseteq [b_{i_k}, d_{i_k})$.

On the other hand, since $d'_{i_k} = d_{i_{k+1}} < d_{i_k} = d'_{i_{k+1}}$, we have that

$$[b'_{i_{k+1}}, d'_{i_{k+1}}) \nsubseteq [b'_{i_k}, d'_{i_k}).$$

However, it respects all other inclusion that $[b_{i_{k+1}}, d_{i_{k+1}})$ respects in the barcode $\mathcal{B}$, i.e.

$$[b'_{i_{k+1}}, d'_{i_{k+1}}) \subseteq [b'_{i_l}, d'_{i_l})$$

for all $0 \leqslant l \leqslant \text{index}_{k+1} - 1$, with $l \neq k$ . Therefore from these inclusions, we see that the bar $[b'_{i_{k+1}}, d'_{i_{k+1}})$ is contained in $(\text{index}_{k+1} - 1)$-bars. This leads to

$$\text{index}_{i_{k+1}}(B') = \text{index}_{i_{k+1}}(B) - 1. \tag{4.3}$$

Moreover, since $b'_{i_k} < b'_{i_{k+1}}$ and $d'_{i_k} < d'_{i_{k+1}}$, the inclusion $[b'_{i_k}, d'_{i_k}) \nsubseteq [b'_{i_{k+1}}, d'_{i_{k+1}})$ doesn't hold either, however is still respects the same inclusion that $[b_{i_k}, d_{i_k})$ respects, i.e for $0 \leqslant l \leqslant \text{index}_k$

$$[b_{i_k}, d_{i_k}) \subseteq [b_{i_l}, d_{i_l}).$$

Hence,

$$\text{index}_{i_k}(B') = \text{index}_{i_k}(B).$$

Since no other bar is affected when going from $B$ to $B'$ we can plug Equation 4.3 in the definition of the $TRN$ for the calculation of $TRN(B')$ .

$$TRN(B') = \prod_{i=1}^{n} \text{index}_i(B') = (\text{index}_{i_{k+1}}(B) - 1) \cdot \prod_{i=1, i \neq k+1}^{n} \text{index}_i(B) =$$

$$= (\text{index}_{i_{k+1}}(B) - 1) \cdot \frac{TRN(B)}{\text{index}_{i_{k+1}}(B)}$$

This calculation yields the result we were looking for. □

**Example 4.32** *To correctly compute $TRN(B'')$, we need to compute the indices of each bar in $B''$*

$$index_1(B'') = 1, \quad index_2(B'') = 1, \quad index_3(B'') = 1,$$
$$index_4(B'') = 2, \quad index_5(B'') = 1.$$

*Note that $B''$ is the barcode coming from the transposition of the deaths $d_3$ and $d_5$ of the barcode $B'$, which has assigned permutation $\rho_{B'} = (41235)$, therefore we are in case 2 of Proposition 4.31. Making now use of Proposition 4.26, we calculate that*

$$TRN(B'') = 24 = \frac{18 \cdot 4}{3} = \frac{TRN(B')(index_5(B') + 1)}{index_5(B')}.$$

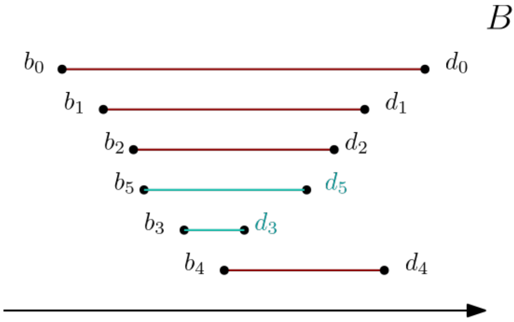


**Figure 4.12:** Transposition of deaths $d_3$ and $d_5$ in barcode $B'$ given in Figure 4.11 used in Example 4.32.

## 4.5 TMD ∘ TNS

Our focus now lies on the theoretical aspects of the composition of TMD with TNS. We want to measure the similarity of barcodes with respect to the bottleneck distance and study the probability of the alteration of two specific bars after applying TMD ∘ TNS. These two aspects together establish stability for the TNS and show that the TNS is a good approximation for a right inverse of the TMD.

### 4.5.1 Similarity of Barcodes

Let $B = \{[b_i, d_i)\}_{0 \leqslant i \leqslant n} \in \mathcal{B}_n^{st}$ be a strict barcode, we fix an appropriate $\lambda$ for the TNS and denote the tree $T_B = TNS(B)$. Now we apply TMD to the tree $T_B$ getting the barcode $B' = TMD(T_B) = \{[b_i', d_i')\}_{0 \leqslant i \leqslant n}$.

From the TNS algorithm, we know that coming close to a new birth or a death in a bar raises the probability of bifurcation respectively termination of the growing branch. Thus for a barcode $B = \{[b_i, d_i)\}_{0 \leqslant i \leqslant n} \in \mathcal{B}_n^{st}$ the distance between a bar $[b_i, d_i) \in B$ and a bar $[b_i', d_i') \in B' = TMD \circ TNS(B) \in \mathcal{B}_n^{st}$ follows an exponential distribution for a fix parameter $\lambda$

$$|b_i - b_i'| \sim \mathrm{Exp}(\lambda) \text{ and } |d_i - d_i'| \sim \mathrm{Exp}(\lambda). \tag{4.4}$$

**Proposition 4.33** *Let $B = \{[b_i, d_i)\}_{0 \leqslant i \leqslant n} \in \mathcal{B}_n^{st}$ and let $B' = TMD \circ TNS(B)$. If $B \sim B'$, then*

$$\mathbb{P}(\tilde{d}_\infty(B, B') > \varepsilon) \leqslant 1 - (1 - e^{-\lambda\varepsilon}(\lambda\varepsilon + 1)^n). \tag{4.5}$$

**Proof** Fix $\gamma \in \mathcal{S}^n$ to be the identity permutation, then

$$\tilde{d}_\infty(B, B') = \sup_{0 \leqslant i \leqslant n} |b_i - b_i'| + |d_i - d_i'|.$$

Since we assume that $B \sim B'$ it directly follows that the identities in Equation 4.4 hold. For $\varepsilon > 0$ the probability that $|b_i - b_i'| + |d_i - d_i'| \leqslant \varepsilon$ is given by

$$\mathbb{P}(|b_i - b_i'| + |d_i - d_i'| \leqslant \varepsilon) = 1 - (1 + \lambda\varepsilon)e^{-\lambda\varepsilon}, \tag{4.6}$$

since the distance between the new values follows an exponential distribution. If we compare Equation 4.6 with the cumulative distribution function of the Erlang distribution given by

$$\mathbb{P}(k, \lambda x) = 1 - \sum_{n=0}^{k-1} \frac{(\lambda x)^n e^{-\lambda x}}{n!},$$

we recognize that the probability distribution function of $|b_i - b_i'| + |d_i - d_i'|$ follows an Erlang distribution with $k = 2$ and $\lambda x = \lambda\varepsilon$. Since the random variables $|b_i - b_i'| + |d_i - d_i'|$ are i.i.d (independent and identically distributed) for all $i \in \{0, \dots, n\}$, considering the sum of $|b_i - b_i'| + |d_i - d_i'|$ over all i leads us to

$$\mathbb{P}(\tilde{d}_\infty(B, B') \leqslant \varepsilon) \geqslant \mathbb{P}(\sup_{0 \leqslant i \leqslant n} |b_i - b_i'| + |d_i - d_i'| \leqslant \varepsilon) = (1 - (1 + \lambda\varepsilon)e^{-\lambda\varepsilon})^n,$$

which after taking the complement is the result from Equation 4.5 that we wanted to achieve. □

Proposition 4.33 is the key for the stability of the TNS with respect to the modified bottleneck distance, depending on the choice of $\lambda$.

### 4.5.2 Transposition of Death Times

Since the TNS is a stochastic process, the computation $TMD \circ TNS(B) = B'$ of a barcode $B \in \mathcal{B}_n^{st}$ and the initial input barcode $B$ are rarely identical, so there might show up some differences. We are therefore interested in the appearance of major changes, making the barcodes $B$ and $B'$ not equivalent. For example, we want to examine if the orders of death times of two bars in $B = \{[b_i, d_i)\}_{i\in\{0,\ldots,n\}}$ changes after applying $TMD \circ TNS$, which would cause the barcodes $B$ and $TMD \circ TNS(B)$ not to be equivalent.

**Proposition 4.34** *Let $B \in \mathcal{B}_n^{st}$ and let $[b_i, d_i), [b_j, d_j) \in B$ such that $d_i < d_j$. Let $[b'_i, d'_i), [b'_j, d'_j)$ be the corresponding bars in $B' = TMD \circ TNS(B)$. The probability that $d'_j < d'_i$ is*

$$\mathbb{P}(d'_j < d'_i) = \frac{1}{2}e^{-\lambda(d_j - d_i)}. \tag{4.7}$$

**Proof** We compute $\mathbb{P}(d'_j < d'_i) = \mathbb{P}(d'_j < d'_i \mid d_i < d_j)$ is the conditional probability of $d'_j < d'_i$ given $d_i < d_j$. We observe that

$$\begin{aligned}\mathbb{P}(d'_j < d'_i) &= \mathbb{P}(d'_j + (d_i + d_j) < d'_i + (d_i + d_j)) = \\ &= \mathbb{P}(d_j + (d_i - d'_i) < d_i + (d_j - d'_j)) = \\ &= \mathbb{P}(d_j + X_i < d_i + X_j) = \mathbb{P}(X_j - X_i < d_j - d_i).\end{aligned}$$

We know that both $X_j$ and $X_i$ follow an exponential distribution with parameter $\lambda$, therefore, defining $Y = X_j - X_i$ leads us to the density function

$$f_Y(t) = \frac{\lambda}{2}e^{-\lambda t}$$

for $t \geqslant 0$, from which the result follows:

$$\mathbb{P}(d'_j < d'_i) = \mathbb{P}(X_j - X_i < d_j - d_i) = \int_{d_j - d_i}^{\infty} f_Y(t)dt = \frac{1}{2}e^{-\lambda(d_j - d_i)}. \qquad \square$$

**Remark 4.35** *Note that there might also occur some birth switches from a barcode $B$ to $TMD \circ TNS(B)$, but we do not analyze them, since the neurons we are interested in have enough distance between the births of their branches.*

Thanks to this powerful proposition, we see that the transposition of two bars is a rare event, which might come up only when the distance between two deaths is very small. This follows, since from Proposition 4.34 we see that the probability of the transportation of two bars decreases exponentially with the distance between their death times.

## 4.6 Characterization of Neuronal Morphologies

In the paper *A Topological Representation of Branching Neuronal Morphologies* [16] it is shown that the Topological Morphology Descriptor (TMD), introduced in Section 4.2.3, can be used to classify different types of neurons. It is based on the comparison between the barcodes constructed with the TMD algorithm, where each barcode encodes the branching structure of the input tree. The TMD provides enough information to create an unbiased benchmark for the categorization of neurons or general trees. Moreover, it can also characterize and quantify the structural differences between morphological groups.

Persistent barcodes are not enough for capturing differences between neuronal trees, hence we use **unweighted persistent images**, mostly represented using heat maps, to which persistent diagrams are converted. We choose this representation since it allows for a construction of an average image for groups of trees, which is useful for quantifying the differences between tree types. The creation of such an image is possible only if we are able to generate a matrix of pixels, representing the persistent diagram in a vector. For the generation of such a matrix, we use a method that discretizes a sum of Gaussian kernels centered at the points of the persistent diagram.

The experiments done in [16] focus on applying the TMD algorithm to different types of trees. First, they categorize randomly generated trees, which have properties that can be modified at will. Next, they discuss two experiments of a more biological nature. The first experiment involves the analysis of neurons from different species and the second one the study of distinct types of rat cortical pyramidal cells. The results of these experiments demonstrate that the TMD can be used to create an efficient classification of them.

We give a detailed explanation of the experiment involving the distinction of neurons from different species. For a further discussion of the other experiments we refer the reader to [18].

**Experiment**: *Study of Neurons from Different Species*

We want to do a topological comparison between neurons of different species: cats, dragonflies, fruit flies, mice, and rats. The corresponding neurons with their persistent diagrams and barcodes are illustrated in Figure 4.13. One can already suspect by looking at neurons in (a), that each of them belongs to a different species since their geometric shape appears to be different. In (b) the corresponding barcode is pictured and we can see more remarkable differences. In barcodes II, III, and V there are a lot of short bars, in contrast to barcodes I and IV, which have the longest bar in the upper part of the graph and more shorter ones as one goes down the $y$-axis. Another remarkable difference is that barcodes III and V have a void in the middle, which is equivalent to saying that no bars were born or died in that period in contrast to barcodes I and IV, which are dense almost everywhere. Part (c) illustrates persistent diagrams and in (d) we can see the persistent diagrams from (c) together with the unweighted persistent image, which allows us to separate the five species into the group they originally belonged to.

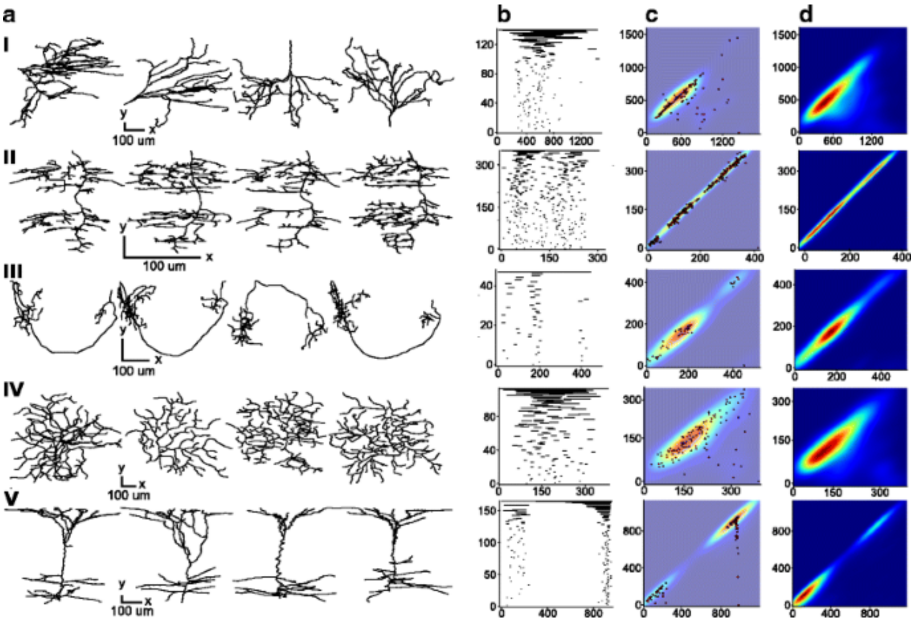


**Figure 4.13:** Part (a) of the figure illustrates neurons for different species, each row corresponds to a species: (I) cat, (II) dragonfly, (III) fruit fly, (IV) mouse, (V) rat. In parts (b) and (c) there are respectively their corresponding persistent barcode and diagram and in (d) we can see an illustration of their unweighted persistent image [16].

What was done in *A Topological Representation of Branching Neuronal Morphologies* [16], mostly covered the beginning of the categorization of rat pyramidal cells. A more in-depth analysis of the morphologies of rat pyramidal cells can be found in the paper *Objective Morphological Classification of Neocortical Pyramidal Cells* [18].

# Bibliography

[1] John M. Bekkers. Pyramidal neurons. *Current Biology*, 21(24):R975, 2011.

[2] Gunnar Carlsson. Topology and data. *Bulletin of The American Mathematical Society*, 46:255–308, 04 2009.

[3] Gunnar Carlsson. Topological pattern recognition for point cloud data. *Acta Numerica*, 23:289–368, 2014.

[4] Gunnar Carlsson and Mikael Vejdemo-Johansson. *Topological data analysis with applications*. Cambridge University Press, 2022.

[5] Frédéric Chazal and Bertrand Michel. An introduction to topological data analysis: fundamental and practical aspects for data scientists, 2021.

[6] Justin Curry, Jordan Deha, Adélie Garin, Kathryn Hess, Lida Kanari, and Brendan Mallery. From trees to barcodes and back again. II: Combinatorial and probabilistic aspects of a topological inverse problem. *Comput. Geom.*, 116, 2024.

[7] Tamal Krishna Dey and Yusu Wang. *Computational topology for data analysis*. Cambridge: Cambridge University Press, 2022.

[8] Reinhard Diestel. *Graph theory*, volume 173 of *Grad. Texts Math.* Berlin: Springer, 5th edition, 2017.

[9] Herbert Edelsbrunner and John L. Harer. *Computational topology. An introduction*. American Mathematical Society (AMS), 2010.

[10] Herbert Edelsbrunner, David Letscher, and Afra Zomorodian. Topological persistence and simplification. *Discrete Comput. Geom.*, 28(4):511–533, 2002.

[11] Ulderico Fugacci, Sara Scaramuccia, Federico Iuricich, and Leila De Floriani. Persistent homology: a step-by-step introduction for newcomers, 10 2016.

[12] Adélie Eliane Garin. From trees to barcodes and back again: A combinatorial, probabilistic and geometric study of a topological inverse problem, 2022.

[13] Peter Giblin. *Graphs, surfaces and homology*. Cambridge University Press, Cambridge, third edition, 2010.

[14] Allen Hatcher. *Algebraic topology*. Cambridge University Press, 2002.

[15] Lida Kanari, Hugo Dictus, Athanassia Chalimourda, Alexis Arnaudon, Werner Van Geit, Benoit Coste, Julian Shillcock, Kathryn Hess, and Henry Markram. Computational synthesis of cortical dendritic morphologies. *Cell Reports*, 39, 2022.

[16] Lida Kanari, Paweł Dłotko, Martina Scolamiero, and et al. A topological representation of branching neuronal morphologies. *Neuroinformatics*, 16:3–13, 2017.

[17] Lida Kanari, Adélie Garin, and Kathryn Hess. From trees to barcodes and back again: theoretical and statistical perspectives, 2020.

[18] Lida Kanari, Srikanth Ramaswamy, Ying Shi, Sebastien Morand, Julie Meystre, Rodrigo Perin, Marwan Abdellah, Yun Wang, Kathryn Hess, and Henry Markram. Objective Morphological Classification of Neocortical Pyramidal Cells. *Cerebral Cortex*, 29(4):1719–1735, 01 2019.

[19] James R. Munkres. *Elements of algebraic topology*. Addison-Wesley Publishing Company, Menlo Park, CA, 1984.

[20] National Institute of Neurological Disorders and Stroke. Brain basics: Know your brain. `https://www.ninds.nih.gov/health-information/public-education/brain-basics/brain-basics-know-your-brain`, 2022. Accessed: 25.06.2024.

[21] Nina Otter, Mason A Porter, Ulrike Tillmann, Peter Grindrod, and Heather A Harrington. A roadmap for the computation of persistent homology. *EPJ Data Science*, 6(1), aug 2017.

[22] Nelson Spruston. Pyramidal neurons: dendritic structure and synaptic integration. *Nature reviews. Neuroscience*, 9:206–221, 04 2008.

[23] Chad M. Topaz, Lori Ziegelmeier, and Tom Halverson. Topological data analysis of biological aggregation models. *PLOS ONE*, 10(5):1–26, 05 2015.

[24] Alan Woodruff. Axons: the cable transmission of neurons. `https://qbi.uq.edu.au/brain/brain-anatomy/axons-cable-transmission-neurons`, 2022. Accessed: 25.06.2024.

[25] Alan Woodruff. What is a neuron? `https://qbi.uq.edu.au/brain/brain-anatomy/what-neuron`, 2022. Accessed: 25.06.2024.

[26] Afra Zomorodian and Gunnar Carlsson. Computing persistent homology. *Discrete Comput. Geom.*, 33(2):249–274, 2005.

[27] Afra J. Zomorodian. *Topology for Computing*. Cambridge Monographs on Applied and Computational Mathematics. Cambridge University Press, 2005.

ETH
Eidgenössische Technische Hochschule Zürich
Swiss Federal Institute of Technology Zurich

## Declaration of originality

The signed declaration of originality is a component of every written paper or thesis authored during the course of studies. In consultation with the supervisor, one of the following three options must be selected:

● I confirm that I authored the work in question independently and in my own words, i.e. that no one helped me to author it. Suggestions from the supervisor regarding language and content are excepted. I used no generative artificial intelligence technologies[1].

○ I confirm that I authored the work in question independently and in my own words, i.e. that no one helped me to author it. Suggestions from the supervisor regarding language and content are excepted. I used and cited generative artificial intelligence technologies[2].

○ I confirm that I authored the work in question independently and in my own words, i.e. that no one helped me to author it. Suggestions from the supervisor regarding language and content are excepted. I used generative artificial intelligence technologies[3]. In consultation with the supervisor, I did not cite them.

**Title of paper or thesis**:

Persistent Homology and Classification of Neuronal Morphologies

**Authored by**:
*If the work was compiled in a group, the names of all authors are required.*

| **Last name(s):** | **First name(s):** |
|---|---|
| Daniele | Melissa |
| | |
| | |
| | |

With my signature I confirm the following:
- I have adhered to the rules set out in the Citation Guide.
- I have documented all methods, data and processes truthfully and fully.
- I have mentioned all persons who were significant facilitators of the work.

I am aware that the work may be screened electronically for originality.

| **Place, date** | **Signature(s)** |
|---|---|
| 9477 Trübbach, 17.7.2024 | Daniele Melissa |
| | |
| | |
| | |

*If the work was compiled in a group, the names of all authors are required. Through their signatures they vouch jointly for the entire content of the written work.*

---

[1] E.g. ChatGPT, DALL E 2, Google Bard
[2] E.g. ChatGPT, DALL E 2, Google Bard
[3] E.g. ChatGPT, DALL E 2, Google Bard