Thèse n° 9852

From Trees to Barcodes and Back Again:A Combinatorial, Probabilistic and Geometric Study of a Topological Inverse Problem

Présentée le 1er avril 2022

Faculté des sciences de la vie
Laboratoire pour la topologie et les neurosciences
Programme doctoral en mathématiques

pour l'obtention du grade de Docteur ès Sciences

par

Adélie Eliane GARIN

Acceptée sur proposition du jury

Prof. A. C. Davison, président du jury
Prof. K. Hess Bellwald, directrice de thèse
Prof. J. Brodzki, rapporteur
Prof. C. Hirsch, rapporteur
Prof. S. Olhede, rapporteuse

École
polytechnique
fédérale
de Lausanne

2022

To all the strong and inspirational
women out there.

Acknowledgements

I would like to start by thanking my supervisor Kathryn Hess for accepting me as a PhD candidate. I am extremely grateful for all your support and the freedom that you gave me to explore the topics I was interested in.

Thank you to the members of my jury, Jacek Brodzki, Christian Hirsch, Sofia Olhede, and to the president, Anthony Davison.

Thank you to Jérôme Scherer, it was real pleasure to share part of the Euler adventure with you. Thank you for always making yourself available and for the nice on and off-work discussions.

To Maroussia, without whom we would all be lost in a nowhere-land, and with whom sharing coffees, meals and chit-chats between doors has always been a pleasure.

To the present and former members of the (extended) group, especially the Top-App subgroup, Jacob Bamberger, Nicolas Berkouk, Stefania Ebli, Daniela Egas Santander, Jeanne Fernandez, Celia Hacker, Darrick Lee, Umberto Lupo, Kelly Maggs, Nicolas Ninin, Azélie Picot, Raphael Reinauer, Erika Roldan, Gard Spreemann, Guillaume Tauzin, you were all incredible colleagues and I will really miss the EPFL team going forward.

A special thank to Gard, Daniela and Guillaume for their help and support at the beginning of this journey, as well as to my fellow PhD students Aras, Celia, Kelly, Haoqing, Lyne, Olivia, Samuel and Stefania.

Thank you to Kate Turner and Vanessa Robins for welcoming me so warmly in Canberra for a month.

I have been very lucky to work with amazing people around the world. Thanks to Lida Kanari, for your patience while explaining that birth and death times are sometimes reversed, to Justin Curry, Jordan DeSha and Brendan Mallery, with whom I have literally spent the most time with on zoom ever, Benjamin Brück, for the hours spent trying to figure out the sense (both literal and figurative) of permutations, to my WinCompTop group members, Teresa Heiss, Brittany Story, Sarah Tymochko, Hoa Bui, Bea Bleile and Vanessa Robins, for the great times spent in Australia, Austria and Skype, and to the interns of EPFL, Ettore Gran, Jeanne Fernandez, Azélie Picot and Florian Richter, it has been a pleasure working with you.

Countless of people have been by my side these past years, without all of whom this work would not have been possible, but I would like to thank a special few. From the math department here and around the world, thanks to (this list is not exhaustive) Emile, Eva, Léo, Florence, Guillaume, Julian, Naya, Paride, Riccardo, Samuel for sharing sat breaks, climate strikes, cookie breaks, travels, conferences, and aperos, amongst other things.

A special thanks to Celia, Darrick, Kelly, Nicolas, Stefania, for being so much more than amazing colleagues, I would never have made it through the covid situation without you.

À Celia, avec qui j'ai partagé bien plus que quatre années de doctorat, qui m'a stimulée et accompagnée dans mes épopées entre autres féministe, écolo et handmade. J'ai hâte de la suite de nos aventures, en van, en podcast ou en agriculture à petite échelle.

À Marie, si forte et si drôle, sans qui ces années de maths n'auraient pas eu la même valeur, tout le meilleur pour la suite de ta double vie.

A Nathalie, à nos 24 ans de vie commune, et aux 75 prochaines années.

Aux colocs de la Lucerna, avec qui j'ai partagé bien plus qu'une incroyable maison. En particulier: Marion, qui m'a apporté un horaire militaire pour aller à l'EPFL, Anil, pour avoir fait de moi une vraie lausannoise, Léonie, qui accepte des idées saugrenues comme aller se baigner à 4h du matin en avril, Matthieu, avec qui partager une conversation à des heures improbables n'a jamais été un problème, et Corinne et Fiona, avec qui j'espère continuer de partager discussions, le jardin, grèves des femmes, apéros et activitiés manuelles. Ma vie à Lausanne n'aurait pas été la même sans vous.

À Simone, pour sa patience et sa capacité à rattraper toutes mes bêtises.

À "les Nous & cie", qui ont cru en moi dès le début, à Chloé, traverser une forêt, un balcon de l'EPFL ou une rue pour te voir est toujours un plaisir, à Jeremy, parce que partager un bateau ce n'est quand même pas rien, à Teo, voici une preuve que c'est quand même moi qui le méritais le plus ce prix.

À mon père, qui serait sans doute très fier de moi. À mes grands-parents et à ma famille, qui ont toujours été là pour moi.

À mon frère Dorian, dont je suis si fière du chemin comme humain, futur médecin et homme déconstruit, à Jean-Pierre, qui, depuis toutes ces années, est un pilier de sagesse et de gentillesse pour nous tous, et à ma mère, Marie-Claude, la personne la plus forte, gentille et humble que je connaisse et sans qui je ne serai pas là aujourd'hui. Merci pour votre soutien inconditionnel.

Last but not least, à Forrest et son incroyable soutien, son ouverture d'esprit, et même son aide pour cette thèse. May you always live on Saturn, and I with you!

Adélie

Abstract

In this thesis, we investigate the inverse problem of trees and barcodes from a combinatorial, geometric, probabilistic and statistical point of view.

Computing the persistent homology of a merge tree yields a barcode B. Reconstructing a tree from B involves gluing the branches back together. We are able to define combinatorial equivalence classes of merge trees and barcodes that allow us to completely solve this inverse problem. A barcode can be associated with an element in the symmetric group Sym_n, and the number of trees with the same barcode, the *tree realization number*, depends only on the permutation type.

We compare these combinatorial definitions of barcodes and trees to those of phylogenetic trees, thus describing the subtle differences between these spaces. The result is a clear combinatorial distinction between the phylogenetic tree space and the merge tree space.

The representation of a barcode by a permutation not only gives a formula for the tree realization number, but also opens the door to deeper connections between inverse problems in topological data analysis, group theory, and combinatorics. Based on the combinatorial classes of barcodes, we construct a stratification of the barcode space. We define coordinates that partition the space of barcodes into regions indexed by the averages and the standard deviations of birth and death times and by the permutation type of a barcode. By associating to a barcode the coordinates of its region, we define a new invariant of barcodes. These equivalence classes define a stratification of the space of barcodes with n bars where the strata are indexed by the symmetric group on n letters and its parabolic subgroups.

We study the realization numbers computed from barcodes with uniform permutation type (i.e., drawn from the uniform distribution on the symmetric group) and establish a fundamental null hypothesis for this invariant. We show that the tree realization number can be used as a statistic to distinguish distributions of trees by comparing neuronal trees to random barcode distributions.

Résumé

Dans cette thèse, on étudie le problème inverse des arbres aux codes-barres d'un point de vue combinatoire, géométrique, probabiliste et statistique.

Le calcul de l'homologie persistante d'un arbre retourne un code-barres B. Pour reconstruire un arbre à partir de B, on recolle les branches en suivant une règle simple. On définit des classes d'équivalence combinatoires d'arbres et de codes-barres qui permettent de résoudre complètement ce problème inverse. Un code-barres peut être associé à un élément du groupe symétrique Sym_n, et le nombre d'arbres ayant le même code-barres, le *nombre de réalisation*, ne dépend que du type de permutation.

On compare ces définitions combinatoires de code-barres et d'arbre à celle d'arbre phylogénétique, décrivant ainsi les différences subtiles entre ces espaces. Il en résulte une distinction combinatoire claire entre l'espace des arbres phylogénétiques et l'espace des arbres.

La représentation d'un code-barres par une permutation donne non seulement une formule pour le nombre de réalisation, mais ouvre également la porte à des connexions plus profondes entre les problèmes inverses en analyse topologique des données, en théorie des groupes et en combinatoire. On définit des coordonnées qui divisent l'espace des codes-barres en régions indexées par les moyennes et les écarts types des naissances et des morts et par le type de permutation d'un code-barre. Associer à un code-barres les coordonnées de sa région définit un nouvel invariant de code-barres. Ces classes d'équivalence définissent une stratification de l'espace des codes-barres avec n barres où les strates sont indexées par le groupe symétrique et ses sous-groupes paraboliques.

On étudie le nombre de réalisation de codes-barres de type permutation uniforme (tirés de la distribution uniforme sur le groupe symétrique), établissant une hypothèse nulle fondamentale pour cet invariant. On montre que le nombre de réalisation peut être utilisé comme statistique pour distinguer les distributions d'arbres en comparant les arbres neuronaux à des distributions de codes-barres aléatoires.

Contents

Chapter 1

Introduction

1.1 Motivation

Studying populations of trees

Trees have a nearly universal presence as a structure for organizing relationships between objects. From hierarchical arrangements that are useful in the classification of species, to more immediate geometric applications in modeling neuron morphology [67–69], trees have proved to be an indispensable tool. Different uses of trees require different types of mathematical definitions. The definition of phylogenetic trees, used to study ancestor relations in species for instance, differs from that of merge trees, which represent the connected components in the sublevel sets of a function. Merge trees can also be used to model geometric trees, enabling the study of objects such as rivers, roots, neurons, etc.

The main issue when working with spaces of trees is that they are not Euclidean spaces, making it hard to study their statistics. As a simple example, the "average tree" of a population is not necessarily defined, or when it is, not unique [99, 107]. A standard way to overcome this problem is to define *invariants* of trees by simplifying their structures. In this thesis, we focus on an invariant of merge trees called *barcodes*, which originates from the field of topological data analysis (TDA). Simply put, a merge tree is described by two types of information: the length of its branches and the adjacency relations between the branches. The barcode of a tree forgets about the adjacency and retains only the information given by the length between the tips of the branches and the branching points. Barcodes have been used to study, for instance, populations of trees in neuroscience [67, 68] as well as for roots [39] and plants [76]. They are convenient summaries of trees because the information that is lost, the adjacency relations of the branches, can be recovered simply by "gluing" the branches back together. How to do so falls into the framework of *inverse problems.* In this thesis, we investigate specifically the inverse problem of trees and barcodes from several points of view.

Applications of this work are mainly in neuroscience, in particular the study of neuron morphology. It was shown in [68] that barcodes help distinguish between different types of neurons. In follow-up work [67], Kanari et al. developed an algorithm to reverse-engineer the process of computing barcodes of neurons. The *Topological Neuronal Synthesis* (TNS) algorithm stochastically generates new trees from a barcode. For now, the TNS is used only on barcodes that are computed directly from existing neurons.

The main motivation behind this work is that, with a complete characterization of the problem from trees to barcodes and back again, and with a good understanding of the space of barcodes, one could be able to generate populations of trees that mimic a given set of trees starting from *artificial* barcodes. From that point on, scientists could study the barcodes that represent best their data using knowledge about the space of barcodes and build the preimage of these barcodes to obtain artificial trees that have similar properties.

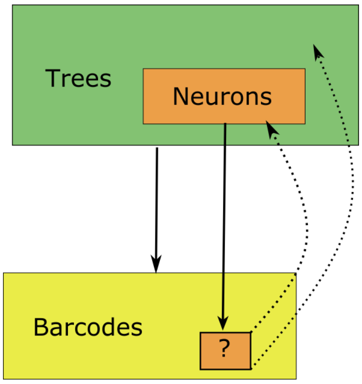


Figure 1.1: Motivation for understanding the preimage of a barcode: Given a barcode computed from a neuron, what do all of its preimages look like?

The space of barcodes

Geometry has been used to analyze data for many years; however, the first topological methods for data analysis were developed only recently, e.g., [23, 44, 54, 101, 104, 108]. At the intersection of data science and algebraic topology, topological data analysis (TDA) is a recent field of study, which provides robust mathematical, statistical and algorithmic methods to analyze the topological and geometric structures underlying complex data. TDA has proved its utility in many applications, including biology [22, 56, 81], material science [72] and climate science [87], and it is still rapidly evolving.

Barcodes are frequently used invariants in TDA. They provide topological summaries of the persistent homology of a filtered space, i.e., a sequence of subspaces $X_t \subseteq X$ of a topological space X included in one another: $X_t \subseteq X_{t'}$ if $t \leq t'$. The barcode $B = \{(b_i, d_i)\}_{i \in \{1,\dots,n\}}$ associated to the filtration $\{X_t\}_{t\in\mathbb{R}}$ is a multiset of points $(b_i, d_i) \in \mathbb{R}^2$ that summarizes the creation and destructions of homology classes throughout the filtration. A bar $(b_i, d_i) \in B$ corresponds to a cycle appearing in X_{b_i} for the first time and becoming a boundary in X_{d_i}. Therefore, the first element of the pair (b_i, d_i) is called the *birth* and the second one the *death*.

In many applications, it is necessary to study statistics on barcodes. Unfortunately, the space of barcodes is not a Hilbert space, which means that it can be difficult to study statistics on it. Several ways to overcome this issue exist, such as the creation of kernels to map barcodes into a Hilbert space [2, 21, 24, 40].

In this thesis, we tackle this issue from a different perspective, by using combinatorial tools from geometric group theory to define new coordinates on the space of barcodes. These coordinates partition the space of barcodes into regions indexed by the averages and the standard deviations of birth and death times and by a permutation associated to a barcode. By associating to a barcode the coordinates of its region, we define a new invariant of barcodes.

Inverse problems

Methods of topological data analysis have been successfully applied in a wide range of fields to provide useful summaries of the structure of complex data sets in terms of topological descriptors, such as barcodes. While there are many powerful techniques for computing topological descriptors, the inverse problem, i.e., recovering the input data from topological descriptors, has proved to be challenging.

Like all summaries, barcodes forget information about the space they are computed from. Thus, even when restricting to a specific set of topological spaces like trees, one may find that many different shapes give rise to the same barcode. Quantifying this failure of injectivity into a summary space is the realm of *topological inverse problems*. Understanding such problems is crucial for comparing different representations of objects arising in both pure mathematics and in data science.

A frequent topic of discussion in the context of TDA is how to define an inverse to the process of associating a particular topological descriptor to a dataset, i.e., how to design a practical algorithm to recover the input data from a topological descriptor, such as a barcode. Oudot and Solomon [92], Leygonie et al. [75] and Curry et al. [31] have proposed partial solutions to this problem. The main obstacle that renders this endeavor particularly challenging has proven to be the computational complexity of the space of inputs considered. To avoid this obstacle, it is reasonable to constrain the input space and search only for an inverse transformation that is relevant in a specific context, for instance, to look for solutions only in the space of embedded graphs, as in [12].

1.2 Contributions

This thesis is a compilation of several published articles and preprints: [69], published in *Algoritms* in collaboration with Lida Kanari and Kathryn Hess, [32] (under review) with Justin Curry, Jordan DeSha, Lida Kanari, Kathryn Hess and Brendan Mallery, [20] in collaboration with Benjamin Brück, and [16, 58], published in the *Young Researcher Forum of SoCG* and *Springer AWM series, special issue on Combinatorial Topology* respectively, in collaboration with Teresa Heiss, Kelly Maggs, Bea Bleile and Vanessa Robins. The chapters of this thesis are re-organized to follow a story-line and not the chronological order in which the results were published. Note that the notation and some definitions differ slightly from one paper to another. For this reason, we summarize the notation and conventions at the beginning of each chapter.

This thesis focuses on the following inverse problem:

$$\begin{tikzcd}
\text{Trees} \arrow[d, shift left, "\text{Persistent Homology}"] \\
\text{Barcodes} \arrow[u, dashed, shift left, "\text{Reconstruction}"] \arrow[d, shift left, "\text{Associated permutation}"] \\
\mathrm{Sym}_n \arrow[u, dashed, shift left, "\text{Stratification}"] \arrow[d, "\text{TRN}"] \\
\mathbb{Z}
\end{tikzcd}$$

Computing the persistent homology of a tree yields a barcode B. To reconstruct a tree from B, one glues the branches back together. We define this inverse problem purely combinatorially, based on combinatorial properties of barcodes: a barcode can be associated with an element in the symmetric group Sym_n, and the number of trees with the same barcode, the *tree realization number* (TRN), depends only on the permutation type. Moreover, each permutation σ represents an equivalence class of barcodes, where the i-th death (in increasing order) is paired with the $\sigma(i)$-th birth (idem). These equivalence classes define a stratification of the space of barcodes with n bars where the strata are indexed by the symmetric group and its *parabolic subgroups.*

The main contributions of this thesis fall into six categories, which form the six following chapters. They also justify the title of this thesis: the first chapter studies the difference between spaces of trees and barcodes, and the three following chapters focus each on a different aspect of the inverse problem: combinatorics, probability and geometric group theory.

The last chapter is self-contained and independent of the rest of this thesis. It focuses on the persistent homology of dual cell complexes.

Delineation of the different spaces of trees and barcodes

The first important outcome of our study is a clear combinatorial distinction between the space of phylogenetic trees (as defined by Billera, Holmes and Vogtmann [13]) and the space of merge trees. Generic combinatorial phylogenetic trees on $n+1$ leaf nodes fall into $(2n-1)!!$ distinct strata, but the analogous number for merge trees is equal to the number of maximal chains in the lattice of partitions, i.e., $(n+1)!n!2^{-n}$. In Chapter 3, we describe the different sets of trees (combinatorial, phylogenetic and merge trees) and the set of barcodes. We explain their main differences and characterize each by defining their combinatorial counterparts. This was joint work with J. Curry, J. DeSha, K. Hess, L. Kanari and B. Mallery, see [32].

Combinatorial characterization of the inverse problem

The general approach to the merge tree-to-barcode inverse problem can be formulated as follows. Any barcode can be realized by finitely many trees, the number of which is called the tree realization number (TRN) or simply the realization number of the barcode. The realization number of a barcode in general position can be computed by certain containment relations between its bars, viewed as intervals on the real line. One crucial observation is that these containment relations partition the set of barcodes (on n bars) into equivalence classes, indexed by permutations in Sym_n. The representation of a barcode by a permutation not only gives a formula for the tree realization number (Lemma 5.13), but also opens the door to deeper connections between inverse problems in TDA, group theory, and combinatorics. We describe a combinatorial characterization of the inverse problem in Chapter 5. This is based on joint work with J. Curry, J. DeSha, K. Hess, L. Kanari and B. Mallery, see [32].

Geometric description of the space of barcodes with n bars

In Chapter 4, we use combinatorial tools from geometric group theory to define new coordinates on the space of barcodes. These coordinates partition the space of barcodes into regions indexed by the averages and the standard deviations of birth and death times and by the permutation type of a barcode. By associating to a barcode the coordinates of its region, we define a new invariant of barcodes. This opens the door to doing statistics on barcodes inspired by the field of permutation statistics. These coordinates define a stratification of the space of barcodes with n bars where the highest dimensional strata are indexed by the symmetric group. This part is based on joint work with B. Brück in [20].

Probabilistic study of the inverse problem

In Chapters 5 and 6 we show that the realization number can be used as a statistic to distinguish distributions of trees. Figure 1.2 shows (log) realization numbers computed from different tree distributions, obtained by computing the realization number either from actual trees, such as neurons, or by randomly generating barcodes with specific properties. The datasets used were *(i)* real neurons (basal and apical dendrites, indicated in red and purple), *(ii)* random barcodes where the birth b_i is picked, then the death d_i is chosen to be larger than b_i, and *(iii)* random barcodes with separated births and deaths so that the induced distribution on the symmetric group is uniform (see Chapter 5). The results are striking: barcodes computed from neurons exhibit a very different distribution than barcodes with uniformly drawn permutation type; see Figure 1.2 for a graphical comparison.

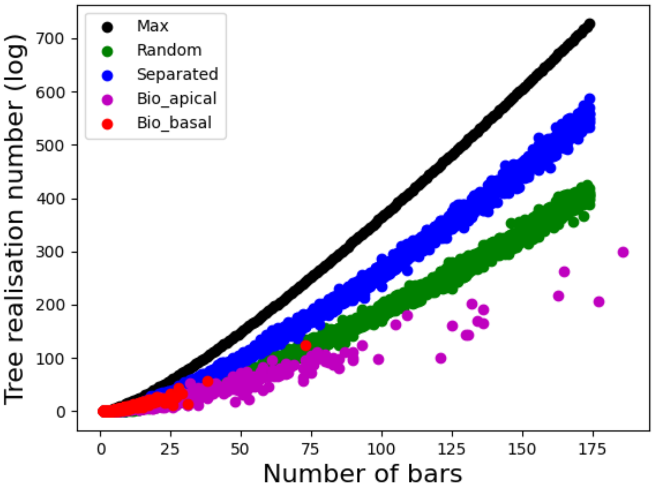


Figure 1.2: The log of the tree-realization number for barcodes with varying numbers of bars for barcodes of basal dendrites (red), apical dendrites (purple) in comparison with “random” barcodes (green), barcodes with separated births and deaths such that the distribution induced on the symmetric group is uniform (blue, see Proposition 5.30), and the maximum tree-realization number ($n!$ for $n+1$ bars) (black).

In this thesis, we study the realization numbers computed from barcodes with uniform permutation type (i.e., drawn from the uniform distribution on the symmetric group). We view this as essential for the realization number to be used for applications, as it establishes a fundamental null hypothesis for the invariant. Our tools are mainly combinatorial, leading us to discover unexpected connections between the inverse problem and other classical combinatorial objects.

Statistical perspectives and stability of a biological inverse problem

We use the TRN as a statistic to study neurons barcodes and the TNS algorithm in Chapter 6. To compute barcodes of neurons, we use the TMD algorithm of [68]. We also investigate stability properties of the TNS. We study the composite of the TNS and TMD algorithms from a theoretical perspective, to quantify the extent to which the TNS acts as an inverse to the TMD. For a given barcode B, we show that, for a reasonable choice of parameter in the TNS, the probability that the bottleneck distance between the barcodes B and $\mathrm{TMD} \circ \mathrm{TNS}(B)$ is greater than ε decreases with ε, thus establishing a form of stability for the TNS. Our stability results imply that the TNS is an excellent approximation to a (right) inverse to the TMD, justifying the use of the TNS to generate artificial neurons [67]. Chapter 6 is joint work K. Hess and L. Kanari see [69].

Duality results in persistent homology

A *filtered complex* is a pair (X, f) of a cell complex X and a cell-wise constant function $f : X \to \mathbb{R}$ such that the sublevel sets of f are subcomplexes. We call two d-dimensional filtered complexes (X, f) and (X^*, f^*) *dual* if (i) each k-dimensional cell $\sigma \in X$ corresponds to a $(d-k)$-dimensional cell $\sigma^* \in X^*$, (ii) the adjacency relations of X are reversed in X^* and (iii) the filtration order is reversed $f^*(\sigma^*) = -f(\sigma)$.

Chapter 7 studies the relationship between the persistent homology of two dual filtered complexes. Our results can be seen as versions or extensions of Alexander duality [86]. We simultaneously generalize existing results for simplicial or polyhedral complexes [47], which were constrained by a number of restrictions, including to spheres (instead of general manifolds) [38, 46], specific functions [46], or standard homology [38]. While our results are similar to those obtained in the study of extended persistence [27], our constructions and proofs differ significantly. We use a pair of dual complexes filtered by complementary functions, whereas [27] uses a single simplicial complex filtered by sublevel and (relative) superlevel sets. Moreover, our results extend to the case of abstract chain complexes derived from discrete Morse theory [52, 88] and refine, for example, the dual V-paths and discrete Morse functions foreshadowed in [11]. This is joint work with K. Maggs, T. Heiss, B. Bleile and V. Robins, see [16, 58].

1.3 Related Work

This thesis touches on many classical concepts related to trees, barcodes, geometric group theory and combinatorics, so the review of the literature done here is not exhaustive.

Inverse Problems in TDA

We review briefly the literature on inverse problems for TDA. The concept of a geometric realization of a persistence module was considered in [73] in order to prove a universality result for the interleaving distance. In [57] the authors initiated an algorithmic study of how to find a point cloud that realizes a given persistence diagram. While these articles are concerned with finding single realizations of persistent signatures, this thesis focuses on the study of the entire pre-image of the persistent homology pipeline.

In the same vein, there is [31], which focused on the setting of functions on the interval and their associated merge trees. Some of the results there were independently discovered and extended in this thesis. The connection between merge trees and discrete Morse functions is studied independently in [19, 65].

More recent articles that investigate the fiber of the persistence map in settings that are different from ours include [35, 64, 75].

We note that the study of the (non-)injectivity of certain topological transforms is also an aspect of topological inverse problems, see [34, 60, 79, 91, 103] for a sampling of these articles and [92] for a recent survey. Better understanding the precise failure of injectivity of certain TDA invariants led to the development of enriched topological summaries that remediate these failures, opening a promising line of research; see [25] and [33] for some examples of these summaries.

Space of Barcodes

The idea of coordinatizing the space of barcodes is not new [40, 66]. For example, the space of barcodes was given tropical coordinates in [66]. In [3], it is mentioned that the space of barcodes can be identified with the n-fold symmetric product of $\mathbb{R}^2$, and the authors study the corresponding algebra of polynomials associated to the variety.

In [105], the author also observes a connection between barcodes and symmetric groups in a different setting, by studying the space of barcode bases using Schubert cells. Similarly, [64] also studies the space of barcode bases.

Spaces of Trees

The wide-spread use of trees in today's research results in countless papers being published on this subject. As such, it is not realistic to have a complete literature review of this large domain, but we do our best to cover the work relevant to this thesis. Defining a polyhedral structure on a space to study statistics has been done for spaces of (phylogenetic) trees [5, 13, 53, 61]. The connection between phylogenetic trees, merge trees and barcodes is studied in Chapter 3. The polyhedral and combinatorial structures defined in this thesis and in [13] or [53] seem to be related, but we leave this as future work (see Section 8.2). Work on statistics of phylogenetic trees include [18, 42, 61, 70, 84]. A lot of work has been done on comparing and studying merge trees, but the space of merge trees is harder to study than that of phylogenetic trees. Defining distances on the set of merge trees and studying geometric properties such as averages, Fréchet means, geodesics has been done in [50, 51, 59, 85, 94, 95, 99, 107]. Comparing merge trees and using them as an invariant has been studied in [77, 96, 106].

Chapter 2

Mathematical Background

This chapter recalls the necessary material for the understanding of this thesis. Basic knowledge of topology, homology and category theory is assumed. The first section covers the notions of graph, which is mainly used to describe tree structures in Section 3.1, and of cell complexes, which take different forms in this thesis: simplicial, cubical or CW complexes. Section 2.3 introduces notions of geometric group theory that are used to describe the barcode space in Chapter 4 and relate it to the number of trees that have the same barcode in Chapters 5 and 6. Section 2.2 reviews the basics of posets and lattices, which are used in Chapter 5 to define the combinatorial inverse problem for trees and barcodes. Lastly, we give a general introduction to the main tool of this thesis, persistent homology, in Section 2.4.

2.1 Graphs and Cell Complexes

In this section, we introduce the basics of graphs and cell complexes. We start with notions about graphs, of which trees form a subset. We then extend the notion of graphs to higher dimensional cell complexes, covering several specific cases: simplicial, cubical, and CW complexes.

2.1.1 Graphs

Graphs can model a wide range of real-world structures: from brain to social networks, human interactions and political opinions, graph theory has been used in many different ways for decades. In this thesis, graphs are mainly used to describe trees structures, which are very specific types of graphs without loops. We begin with basic definitions in graph theory.

Definition 2.1. A *graph* G is a pair (V, E) of sets of *vertices*, V, and *edges* $E \subseteq \mathcal{P}(V)$ such that $|e| = 2$ for any $e \in E$. A graph is *finite* if V and E are finite sets. If $\{v, w\} \in E$ defines an edge, v and w are called *adjacent* vertices. The *degree* of a vertex v is the number of edges that contain v. If the degree is either 1 or 3 for every vertex in G, the graph is said to be *binary.* A *path* in the graph is a sequence of adjacent vertices $v_1, \ldots, v_n$. A *cycle* is a path $v_1, \ldots, v_n$ such that $v_i \neq v_j$ if $i \neq j$ apart from $v_1 = v_n$. A graph that does not contain any cycle is called *acyclic.*

Graphs are sometimes equipped with additional structure. For example, some graphs are *weighted*, that is, there is a map $\omega : E \longrightarrow \mathbb{R}$ assigning a *weight* $\omega(e) \in \mathbb{R}$ to each edge e.

Even though a graph G contains only the information of the set of vertices and the adjacency relations given by the edges, in practice it is easier to index the vertices. A *labelling* of a graph is a map $\mathcal{L} : V \longrightarrow S$ from the vertices to a set of labels S. If S is a subset of the natural numbers $\mathbb{N}$, the labelling is *ordered*.

A graph can be encoded via an *adjacency matrix*, a matrix A indexed by the vertices of the graph, where A_{ij} is the number of edges between vertex v_i and vertex v_j, i.e., 0 or 1. Adjacency matrices are symmetric. Technically speaking, writing down an adjacency matrix of a graph G requires choosing an ordered labelling of the vertices, which can be arbitrary. This creates difficulties in identifying adjacency matrices coming from the same graphs, which brings us to the notion of graph isomorphism.

Definition 2.2. A graph morphism between graphs $G = (V, E)$ and $G' = (V', E')$, denoted by $f : G \longrightarrow G'$, is a map $f : V \longrightarrow V'$ such that if $\{v, w\} \in E$, then $\{f(v), f(w)\} \in E'$. It is a graph *isomorphism* if f is a bijection and induces a bijection between E and E'.

To identify isomorphic graphs, their adjacency matrices turn out to be useful. Two graphs G, G' with corresponding adjacency matrices A, A' are isomorphic if there exists a permutation matrix P (which contains exactly one 1 per row and column, and 0 elsewhere) such that $PAP^{-1} = A'$. In other words, to determine whether two graphs are isomorphic, one has to find ordered labelling of both such that the corresponding adjacency matrices agree.

A graph is *directed* if the pairs of vertices in the set of edges E are ordered, i.e., if $E \subseteq V \times V$. In this case, we say that an edge (v, w) *goes* from v to w. The adjacency matrix of a directed graph is not necessary symmetric anymore, as there can be a directed edge from v to w and none from w to v.

The *graph path distance* is a distance on the vertices of the graph. The distance $d(v, w)$ between two vertices $v, w \in V$ is the minimal number of edges in a path between v and w.

2.1.2 Cell Complexes

We now introduce the notion of cell complexes, which we will mainly use to describe polytopes. Polytopes can be viewed as higher dimensional versions of graphs.

They are geometric objects with “flat” sides. More formally, a *(finite) polytope* is the convex hull of a finite number of points in $\mathbb{R}^n$. There are several important classes of polytopes, of which we study simplicial complexes and cubical complexes in this thesis. We also discuss another type of cell complex, CW complexes, that we use in Chapter 7.

Simplicial Complexes

Simplicial complexes, like graphs, are built from vertices and edges, but also from higher dimensional objects, called k-simplices. They can be defined purely abstractly.

Definition 2.3. A *finite abstract simplicial complex* is a finite set A together with a collection $K \subseteq \mathcal{P}(A)$ of subsets of A such that if $\sigma_1 \in K$ and $\sigma_2 \subseteq \sigma_1$ then $\sigma_2 \in K$. To simplify the notation, the pair (A, K) is usually denoted by K.
If $v \in A$ and $\{v\} \in K$, v is called a *vertex.*
The sets $\sigma \in K$ are called *simplices.* The *dimension* of a simplex σ is $|\sigma| - 1$, and the *dimension* of the complex K is the maximal dimension of its simplices. The *d-skeleton* of K is the set of all simplices of dimension d.

Simplicial complexes can be realized geometrically into geometric simplicial complexes, which we introduce now.

Definition 2.4. A *k-dimensional simplex* or *k-simplex* σ in $\mathbb{R}^n$ is the convex hull of $k+1$ affinely independent points $x_0, ..., x_k$. A 0-simplex is also called a *vertex*, a 1-simplex an *edge*, and a 2-simplex a *face.*
For $m \leq k$, a *m-face* τ of σ, denoted by $\tau \leq \sigma$, is the convex hull of a subset of $m+1$ points of the generating set of σ. If $\tau \neq \sigma$, it is a *proper* face.

Example 2.5. Figure 2.5 shows examples of simplices for $n = 0, 1, 2, 3$. A 0-simplex is a vertex, a 1-simplex is an edge, a 2-simplex is a triangle and a 3-simplex is a tetrahedron.

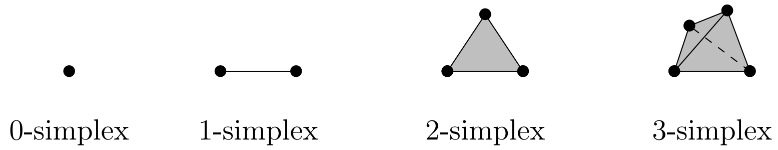


Figure 2.1: Low dimensional simplices for $n = 0, 1, 2, 3$.

Definition 2.6. A *finite geometric simplicial complex* K is a finite collection of simplices satisfying two conditions:

1. For any simplex $\sigma \in K$ and m-face $\tau \leq \sigma$, $\tau \in K$.

2. If σ_1,σ_2 are two simplices of K, then $\sigma_1 \cap \sigma_2$ is either empty or a face of both.

The *dimension* of a geometric simplicial complex is the maximal dimension of its simplices. The d-skeleton of K is the set of all simplices of dimension d.

Note that geometric simplicial complexes are topological spaces as they inherit the ambient topology of the Euclidean space $\mathbb{R}^n$.

The following definition holds for both abstract and geometric simplicial complexes.

Definition 2.7. A finite simplicial complex K_1 is a *subcomplex* of K_2, denoted by $K_1 \subseteq K_2$, if each simplex of K_1 is a simplex of K_2.

Remark 2.8. To describe a finite simplicial complex, one needs only to know its maximal simplices, the ones that are not proper faces of another simplex. The others are all included in the maximal ones as faces, because of the first condition of Definition 2.6.

Remark 2.9. A finite geometric simplicial complex K consists of a set of vertices A, called the *vertex set*, with a subset $K \subseteq \mathcal{P}(A)$ representing the simplices. It can hence be considered as an abstract simplicial complex if one only sees its combinatorics and forgets about the ambient space $\mathbb{R}^n$. On the other hand, any finite abstract simplicial complex K can be associated to a geometric simplicial complex $|K|$ in $\mathbb{R}^n$ for n sufficiently large. To do so, index an affinely independent family of vectors by the set of vertices of K. Then each simplex $\sigma = \{x_0, ...x_n\}$ of K is associated to the convex hull of the corresponding vectors and $|K|$ is the union of all the geometric simplices, equipped with the subspace topology.

Simplicial complexes are one of the most common ways to discretize topological objects. A *triangulation* of a topological space X is a simplicial complex K that is homeomorphic to X. For instance, every closed compact 2-manifold admit a triangulation [41]. Moreover, any polytope admits a simplicial decomposition (this is sometimes even the definition of a polytope).

A simplicial complex is naturally *stratified* by its simplices. Note that with the usual topology on the geometric realization of a simplicial complex, the simplices are closed.

Definition 2.10. [17] A topological space X is *stratified* over a poset $\mathcal{P}$ if there exists a collection of subsets $\{X_i\}_{i \in \mathcal{P}}$ of X, called the *strata*, such that:

1. $X = \bigcup_i X_i$;

2. $i \leq j$ if and only if $X_i \subseteq X_j$;

3. if $X_i \cap X_j \neq \emptyset$, then it is a union of strata;

4. For every $x \in X$, there exists a unique $i_x \in \mathcal{P}$ such that $\bigcap_{x \in X_j} X_j = X_{i_x}$.

The inclusion of strata gives a partial ordering on $\mathcal{P}$: $i \leq j$ if and only if $X_i \subseteq X_j$.

Cubical Complexes

Cubical complexes are another type of high-dimensional polytopes, where the building blocks are cubes instead of triangles.

Definition 2.11. An *elementary interval* is an interval of the form $I = [l, l+1]$ or $[l, l]$. It is *degenerate* if $I = [l, l]$. An *elementary k-cube* $\sigma \subset \mathbb{R}^d$ is a product of d elementary intervals,

$$\sigma = I_1 \times I_2 \times \ldots \times I_d$$

such that $d - k$ of the intervals are degenerate.

A *cubical complex* $X \subset \mathbb{R}^d$ is a cell complex consisting of a set of elementary k-cubes, such that all faces of $\sigma \in X$ are also in X, and such that all vertices of X are related by integer offsets.

CW Complexes

CW-complexes [78] (C for “closure-finite” and W for “weak topology”) generalize simplicial complexes and cubical complexes to allow cells that are not necessarily simplices or cubes but homeomorphic to open disks or balls. They provide a framework for cellular decomposition of topological spaces.

Definition 2.12. A finite *CW complex* X of dimension d is defined recursively by dimension, building the k-skeleton based on the $(k-1)$-skeleton.

1. The 0-skeleton X^0 is just a set of points equipped with the discrete topology.
2. The k-skeleton X^k is built by attaching k-cells (open k-dimensional balls) D^k_α via attaching maps from the boundary S^k_α of the balls to the $(k-1)$-skeleton of X, $\varphi^k_\alpha : S^{k-1} \longrightarrow X^{k-1}$. The k-skeleton X^k is the quotient space
$$X^k = X^{k-1} \sqcup \bigsqcup_\alpha D^k_\alpha / x \sim \varphi_\alpha(x),$$
where $x \in \partial D^k_\alpha \simeq S^{k-1}$ is identified with its image by φ^k_α. The process stops when $k = d$.

A CW-complex X is *regular* if the closure of each k-cell in X is homeomorphic to the closed k-dimensional ball D^k.

Let X be a CW complex and τ and σ two cells of X. If $\tau \subseteq \overline{\sigma}$, then τ is a *face* of σ, and σ is a *coface* of τ, denoted by $\tau \preceq \sigma$. The *codimension* of a pair of cells $\tau \preceq \sigma$ is the difference in dimension, $\dim(\sigma) - \dim(\tau)$. If σ has a face τ of codimension 1, we call τ a *facet* of σ, and write $\tau \lhd \sigma$. A function $f : X \to \mathbb{R}$ on the cells of X is *monotonic* if it increases with the dimension, that is, $f(\tau) \leq f(\sigma)$ whenever $\tau \preceq \sigma$.

Dual cell complexes

The last definition of this section is essential for Chapter 7. It describes dual cell complexes. Intuitively, two cell complexes X and X^* of dimension d are dual if their cells are "inverted": for each cell of dimension k in X there is a corresponding cell of dimension $d-k$ in X^*, and the face relations are also reversed.

Definition 2.13. Two d-dimensional cell complexes X and X^* are *combinatorially dual* if there is a bijection $X \to X^* : \sigma \mapsto \sigma^*$ between the sets of cells such that

1. (Dimension Reversal) $\dim(\sigma^*) = d - \dim \sigma$ for all $\sigma \in X$.
2. (Face Reversal) $\sigma \preceq \tau \iff \tau^* \preceq \sigma^*$ for all $\sigma, \tau \in X$.

This definition has a more combinatorial interpretation that we describe in Section 2.2.1.

2.2 Algebraic Combinatorics

In Chapter 5, we identify combinatorial types of trees with maximal chains in a specific lattice. In this section, we introduce the necessary background on posets and lattices but keep it as self-contained as possible.

2.2.1 Posets

Definition 2.14. A *partially ordered set*, or *poset*, $(P, \leq)$ is a set P with a binary relation $\leq$ satisfying the following conditions for all $p, q, r \in P$:

1. (Reflexivity) $p \leq p$,
2. (Antisymetry) if $p \leq q$ and $q \leq p$ then $p = q$
3. (Transitivity) if $p \leq q$ and $q \leq r$, then $p \leq r$.

If the set P is finite, then the poset is *finite*. Given posets $(P, \leq_P)$ and $(Q, \leq_Q)$, a map $f : P \longrightarrow Q$ is *order preserving* if $p \leq_P q$ implies $f(p) \leq_Q f(q)$ for all $p, q \in P$, and we write $f : (P, \leq_P) \longrightarrow (Q, \leq_Q)$.

Example 2.15. The natural numbers $\mathbb{Z}$ and real numbers $\mathbb{R}$ equipped with the usual order $\leq$ are infinite posets.

In this thesis, all posets are finite, apart from the two examples above and their respective subsets $\mathbb{N}$ and $\mathbb{Q}$.

For any poset $(P, \leq)$, we can specify its *opposite poset* or *dual poset* $(P^{op}, \leq_{op})$ as the poset whose elements are the same as P, with order relation given by $p \leq_{op} q$ whenever $p \geq q$ in P.

One of the most natural examples of a poset is the collection of all subsets of a set X ordered by inclusion. We review this example in the next section.

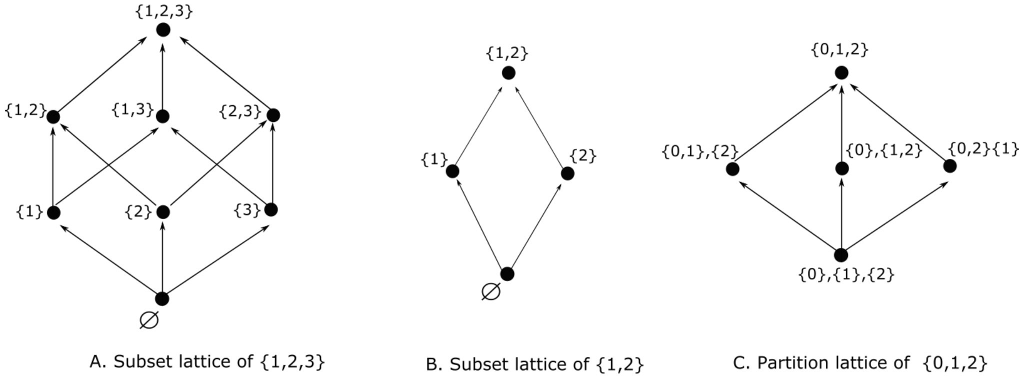
Example 2.16. The faces of a cell complex comprise elements of a poset, serving as a useful combinatorial summary of the relationships between the cells. Let X be a regular cell complex. The *face poset* $\mathsf{Face}(X)$ consists of the set of cells of X with order relation $\tau \leq \sigma$ if and only if $\tau \preceq \sigma$. Moreover, if X and X^* are two dual cell complexes, their face posets are also dual.

We can thus extract a poset from a topological object like a cell complex. In fact, we can go in the other direction - from any poset, we may construct a simplicial complex called the *order complex.* Given any regular cell complex X, it turns out that the order complex of the face poset is the barycentric subdivision of the original decomposition [78] and, hence, homotopy equivalent to X.

A *totally ordered set* is a poset P such that for each $p, q \in P$, either $p \leq q$ or $q \leq p$. If a poset P contains a subset $\mathcal{C} \subseteq P$ that is totally ordered, then $\mathcal{C}$ is called a *chain.* A chain is *maximal* if it is of maximal length in P.

A point $p \in P$ is said to be *covered* by $q \in P$, denoted by $p \precnsim q$, if $p \leq q$ and there is no $v \in P$ such that $p \leq v \leq q$. The partial order in P is completely determined by the covering relation: the order $\leq$ is the smallest reflexive and transitive relation that contains $\precnsim$. In particular, this can be used to define the *Hasse diagram* of a poset $(P, \leq)$. The Hasse diagram H of $(P, \leq)$ is a (directed) graph of which the vertices are the elements of P and such that there is an edge from p to q if $p \precnsim q$.

We show three Hasse diagrams in Figure 2.2.



A. Subset lattice of {1,2,3}

B. Subset lattice of {1,2}

C. Partition lattice of {0,1,2}

Figure 2.2: A. The Hasse diagram of the subset lattice of $\{1,2,3\}$ of Example 2.18 B. Another subset lattice for of $\{1,2\}$. C. The Hasse diagram of the partition lattice of $\{0,1,2\}$ of Example 2.20.

2.2.2 Lattices

The theory of lattices is rich and complex. We will not need the abstract definition of lattice in this thesis but the reader interested in this topic can read [89]. Here, we will think of a lattice as a poset that has a *least upper bound* (supremum) and a *greatest lower bound* (infimum), defined below.

Definition 2.17. Let $(P, \leq)$ be a poset and $Q \subseteq P$. A point $u \in P$ is called an *upper bound* of Q if $q \leq u$ for all $q \in Q$. It is a *least upper bound* if $u \leq w$ for all other upper bounds $w \in Q$. Dually, $l \in P$ is a *lower bound* of Q if $l \leq q$ for all $q \in Q$. It is a *greatest lower bound* if $w \leq l$ for all other lower bound w of Q.

We are mainly interested in two specific lattices that we introduce now.

Example 2.18. [Subset Lattice] Let $[n] = \{1, \ldots, n\}$ and consider $P = \mathcal{P}([n])$, the set of all subsets of $[n]$. Equip P with the partial order $\subseteq$ of “being a subset of”. This forms the *subset lattice* Π_n of $[n]$. Figure 2.2A and B show the Hasse diagram of the subset lattices Π_3 and Π_2 of $\{1,2,3\}$ and $\{1,2\}$ respectively.

Definition 2.19. A *partition* of the set $\mathbf{n} := \{0, 1, \ldots, n\}$ is a collection of pairwise disjoint subsets $\mathcal{U} = \{U_1, \ldots, U_k\}$ of $\mathbf{n}$ whose union is $\mathbf{n}$. A partition $\mathcal{U}$ *refines* a partition $\mathcal{U}'$, written $\mathcal{U} \preceq \mathcal{U}'$, if every subset of $\mathcal{U}'$ is equal to a union of elements of $\mathcal{U}$. Said differently, $\mathcal{U} \preceq \mathcal{U}'$ if for each $U_i \in \mathcal{U}$ there exists $U'_j \in \mathcal{U}'$ such that $U_i \subseteq U'_j$. We denote the set of partitions of $\mathbf{n}$ by $\mathcal{P}_n$. The refinement relation endows the set $\mathcal{P}_n$ with a partial order, which also happens to be a lattice. A chain in the lattice of partitions is a sequence of comparable partitions

$$\mathcal{U}_1 \preceq \cdots \preceq \mathcal{U}_\ell.$$

Such a chain is maximal if it is not a subsequence of any longer chain.

For the sake of notation, we can always write a partition of $\mathbf{n}$ as an ordered list where each subset is separated by a vertical line. The finest possible partition—and hence the bottom element of the $\mathcal{P}_n$—is denoted

$$\{0|1|2|\cdots|n\}.$$

The top element of $\mathcal{P}_n$ is the set $\mathbf{n}$.

Example 2.20. Figure 2.2C shows $\mathcal{P}_2$, the lattice of partitions of $\{0, 1, 2\}$.

2.3 Geometric Group Theory

Geometric group theory is a wide field of mathematics studying the "shape" of groups. In this thesis, we use the symmetric group to identify equivalence classes of barcodes. Chapter 4 goes further into this study and add a geometric structure to the equivalence classes. Many tools from geometric group theory can be used to study barcodes through this identification, and we introduce the necessary background now.

2.3.1 Presentations of Groups

We start with the notion of a presentation of a group G, which is a method to represent G with a set of *generators* S and a set of *relations* R, which is a subset of F_S, the free group generated by S. We denote a presentation by $G = \langle S \mid R\rangle$.

The group G is the quotient of the free group F_S and the smallest normal subgroup N of F_S that contains R:

$$G = \langle S \mid R\rangle = F_S/N.$$

The set S is sometimes called an *alphabet*, and a *word* is a sequence of elements of S or their formal inverses. The set of relations R is a set of words that are equivalent in the quotient. The *length* of an element $g \in G$ is the minimal length of a word representing g. A word representing a certain element is *reduced* if it is of minimal length.

Presentations of groups have a nice graphical reprentation as graphs. The *Cayley graph* of $G = \langle S \mid R\rangle$ is a graph with vertex set G and an edge between g_1 and g_2 if there is an element $s \in S$ such that $g_1 = sg_2$. Figure 3.8 shows an example of the Cayley graph of the symmetric group Sym_4 generated by the adjacent transpositions.

The length of a word corresponds to the shortest path between the word and the identity id on the Cayley graph. Using the conventions above, one can also define a partial order on a group as follows.

Definition 2.21 (Left Bruhat Order)**.** The *left Bruhat order* on a group G with representation $\langle S \mid R \rangle$ is a partial order on G, specified as follows. If $\sigma, \sigma' \in G$, then $\sigma < \sigma'$ if the length of σ is less than that of σ', and there exist $\tau_{i_1}, ..., \tau_{i_k} \in S$ such that $\sigma' = \tau_{i_1} \cdots \tau_{i_k} \sigma$.

One of the most common examples of a group presentation, and the one we are mainly interested in in this thesis, is the usual presentation of the symmetric group Sym_n.

Recall that the symmetric group is generated by elementary transpositions $\tau_i = (i, i+1)$. This implies that any element of Sym_n can be represented using a word with the alphabet $S = \{\tau_i\}_{i=1}^{n-1}$, although that representation will not be unique.

Remark 2.22 (Different Notation for Permutations). There are several notational conventions for elements of the symmetric group. When we use square brackets or boxes, e.g., the notation [132], then we are listing the images of the ordered set $\{1, \ldots, n\}$ under the map σ, e.g., for $\sigma = [132]$, one can read off that $\sigma(1) = 1$, $\sigma(2) = 3$ and $\sigma(3) = 2$. We also use cycle notation, which describes the permutation in terms of its orbits and uses parentheses; fixed points are omitted in this notation. For our example, $\sigma = [132]$ can also be written as the elementary transposition (23).

2.3.2 Coxeter System and Coxeter Complex

Coxeter groups

Coxeter groups form a family of groups that was defined by Tits in its modern form. They are abstract versions of reflection groups; in fact, the family of finite Coxeter groups coincides with the family of finite reflection groups. Besides their close connections to geometry and topology [36], Coxeter groups have a rich combinatorial theory [15]. They appear in many areas of mathematics, e.g. as Weyl groups in Lie theory. We will view Sym_n as one of the most basic examples of a Coxeter group.

Usually, one does not consider a Coxeter group W by itself but instead a *Coxeter system* (W, S), where S is a generating set of W that consists of involutions called the *simple reflections*. In what follows, we will tacitly assume that such a set of simple reflections is always fixed when we talk about a Coxeter group W. In the case where $W = \mathrm{Sym}_n$, we will take S to be the set of adjacent transpositions $S = \{(i, i+1) \mid 1 \leq i \leq n-1\}$. A rank-$(|S| - 1 - k)$ (standard) *parabolic subgroup* of W is a subgroup of the form $P_T = \langle T \rangle$, where $T \subset S$ is a subset of size $(|S| - 1 - k)$.

Coxeter complexes

Each Coxeter group W can be assigned a simplicial complex $\Sigma(W)$, the *Coxeter complex*, that is equipped with an action of W. If W is a finite group with set of simple reflections S, the complex $\Sigma(W)$ is a triangulation of a sphere of dimension $|S|-1$. Coxeter complexes have nice combinatorial properties and are in particular colourable flag complexes [1, Section 1.6] that are shellable [14].

The top-dimensional simplices of $\Sigma(W)$ are in one-to-one correspondence with the elements of the group W. Furthermore, one recovers the Cayley graph of (W, S) as the *chamber graph* of $\Sigma(W)$, i.e., the graph that has a vertex for each top-dimensional simplex of $\Sigma(W)$ and an edge connecting two vertices if the corresponding simplices share a codimension-1 face [1, Corollary 1.75].

More generally, the set of k-simplices in $\Sigma(W)$ is in one-to-one correspondence with the cosets of rank-$(|S| - 1 - k)$ parabolic subgroups of W:

Definition 2.23. The *Coxeter complex* $\Sigma(W)$ of the Coxeter system (W, S) is the simplicial complex

$$\Sigma(W) = \bigcup_{T\subseteq S} W/P_T = \{\tau P_T \mid \tau \in W, T \subseteq S\},$$

where the simplex τP_T has dimension[1] $\dim(\tau P_T) = |S \setminus T| - 1$ and the face relation is defined by the partial order

$$\tau P_T \leq \tau' P_{T'} \Leftrightarrow \tau P_T \supseteq \tau' P_{T'}. \tag{2.1}$$

The group W acts simplicially on $\Sigma(W)$ by left multiplication on the cosets, $\gamma \cdot (\tau P) := \gamma\tau P$.

Remark 2.24. With a slight abuse of notation, we will in what follows often use the cosets τP to also denote simplices in the geometric realization of the Coxeter complex. To be coherent with the definition of a stratification (Theorem 2.10), we will always consider these simplices to be closed.

[1] Note that we take the (combinatorial) convention that this simplicial complex has a unique face of dimension -1. This face does not appear in the geometric realization.

The Coxeter complex $\Sigma(\mathrm{Sym}_n)$

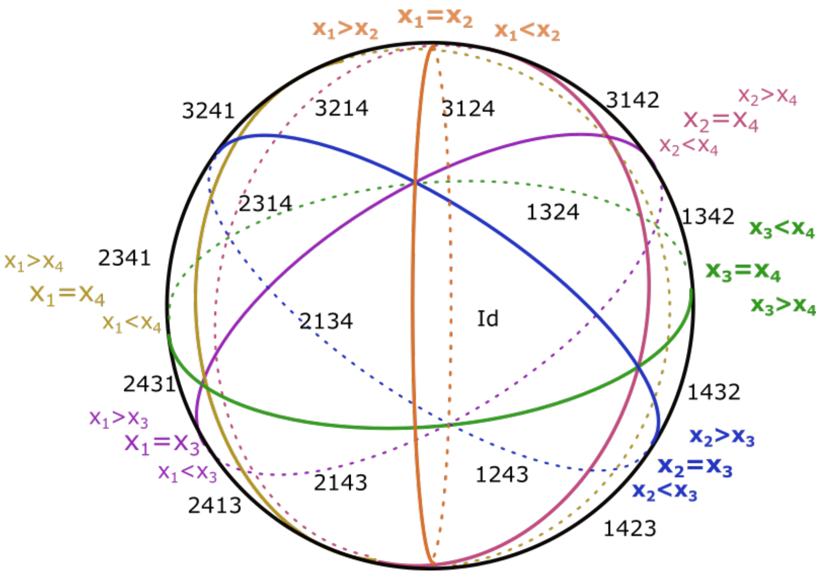


Figure 2.3: The geometric realization of the Coxeter complex $\Sigma(\mathrm{Sym}_4)$. The permutation corresponding to each triangle of the front of the sphere is indicated in black. The hyperplanes $x_i = x_j$ depicted in colours correspond to the transpositions (i, j). The hyperplanes corresponding to adjacent transpositions $(i, i+1)$ are in boldface. A detailed description of how to obtain such a geometric realization of the Coxeter complex can be found in Section 4.2.1.

For the case $W = \mathrm{Sym}_n$ that we are interested in, the Coxeter complex $\Sigma(\mathrm{Sym}_n)$ is of dimension $n-2$ and is isomorphic to the barycentric subdivision of the boundary of an $(n-1)$-simplex. It can be realized geometrically as a triangulation of the $(n-2)$-sphere. This complex is the dual to the permutohedron of order n (see Figure 4.5). Figure 2.3 depicts the Coxeter complex $\Sigma(\mathrm{Sym}_4)$. The top-dimensional simplices of $\Sigma(\mathrm{Sym}_n)$ are in one-to-one correspondence with the elements of Sym_n. Two such simplices share a codimension-1 face if and only if the corresponding permutations differ by precomposing with an adjacent transposition $(i, i+1)$, i.e., by exchanging two neighbouring entries of the permutation. As a consequence, if x lies in the interior of a maximal simplex of the geometric realization of $\Sigma(\mathrm{Sym}_n)$, it can be assigned a permutation $\tau \in \mathrm{Sym}_n$. If x lies on a face of dimension k, then τ is well-defined only up to multiplying by an element of a parabolic subgroup $P \leq \mathrm{Sym}_n$ that is generated by $|S| - 1 - k = n - 2 - k$ adjacent transpositions. A concrete embedding of $\Sigma(\mathrm{Sym}_n)$ in $\mathbb{R}^n$ will be described in more detail in Section 4.2.1.

For later reference, we note that the identification $S^{n-2} \cong \Sigma(\mathrm{Sym}_n)$ gives a *stratification* of the sphere by its simplicial decomposition. The strata are the (closed) simplices of the geometric realization and the stratification is over the partially ordered set (poset) specified by Equation 2.1.

2.4 Persistent Homology

In this section, we describe one of the main tool of this thesis, persistent homology. We give a general introduction here, even though we use a more combinatorial definition in most of this thesis. Persistent homology is an invariant of a sequence of subspaces of a fixed space ordered by inclusion. It is a finer invariant than the homology of a space X, as it encodes the homology at each step of a sequence of subspaces ending in X and the inclusion maps in between. When working with data, standard topological quantities such as homology can be highly sensitive to noise and small geometric fluctuations. Persistent homology addresses this problem by examining a collection of spaces, indexed by a real variable often representing an increasing length scale. These spaces are modelled by a cell complex X with a filter function $f : X \to \mathbb{R}$ assigning to each cell the scale at which this cell appears. This process is called a filtration.

2.4.1 Filtrations

Definition 2.25. A filtration of a topological space X is a set of subspaces $\{X_t\}_{t\in\mathbb{R}}$ such that $X_t \subseteq X_{t'} \subseteq X$ if $t \leq t'$. The set $\{X_t\}_{t\in\mathbb{R}}$ is also called a *filtered space.*

A filtration $\{X_t\}_{t\in\mathbb{R}}$ can come from a map $f : X \longrightarrow \mathbb{R}$ via its *sublevel sets* $X_t = f^{-1}((-\infty, t])$. The parenthesis are sometimes omitted to avoid heavy notation. A function $f : X \longrightarrow \mathbb{R}$ is *tame* if the homology groups of its sublevel sets have finite rank and change at a finite number of $t \in \mathbb{R}$.

When X is a cell complex, the function f is required to be monotonic so that each sublevel set is a cell complex, leading to the following definition.

Filtrations of cell complexes

Definition 2.26. A *filtered (cell) complex* (X, f) is a cell complex X together with a monotonic function $f : X \to \mathbb{R}$. A linear ordering $\sigma_0, \sigma_1, \ldots, \sigma_n$ of the cells in X, such that $\sigma_i \preceq \sigma_j$ implies $i \leq j$, is *compatible* with the function f when

$$f(\sigma_0) \leq f(\sigma_1) \leq \ldots \leq f(\sigma_n).$$

Note that the monotonicity condition implies that, for $r \in \mathbb{R}$, the sublevel set

$$X_t = f^{-1}(-\infty, t]$$

is a subcomplex of X. The value $f(\sigma)$ determines when a cell enters the filtration given by this nested sequence of subcomplexes. The definition of a compatible ordering also implies that each step in the sequence

$$\emptyset \subset \{ \sigma_0 \} \subset \{ \sigma_0, \sigma_1 \} \subset \cdots \subset \{ \sigma_0, \sigma_1, \ldots, \sigma_n \}$$

is a subcomplex, and every sublevel set $f^{-1}(-\infty, t]$ appears somewhere in this sequence: $f^{-1}(-\infty, t] = f^{-1}(-\infty, f(\sigma_i)] = \{ \sigma_0, \sigma_1, \ldots, \sigma_i \}$ for $i = \max\{ i = 0, \ldots, n \mid f(\sigma_i) \leq t \}$.

Dual Filtrations

We have already seen the notion of dual cell complexes in Definition 2.13. This definition can be extended to dual filtered complexes, where the duality is compatible at each step of the filtration functions. The next definition and proposition are necessary only for Chapter 7.

Definition 2.27. Two filtered complexes (X, f) and (X^*, g) are *dual filtered complexes* if X and X^* are combinatorially dual to one another and if there exists a linear ordering $\sigma_0, \sigma_1, \ldots, \sigma_n$ of the cells in X that is compatible with f and such that its dual ordering $\sigma_n^*, \sigma_{n-1}^*, \ldots, \sigma_0^*$ is compatible with g.

The following lemma gives a simple condition under which the filtration functions of (X, f) and (X^*, f^*) are dual filtered complexes.

Lemma 2.28. *Suppose two functions $f : X \to \mathbb{R}$ and $f^* : X^* \to \mathbb{R}$ satisfy $f^*(\sigma^*) = -f(\sigma)$. Then (X, f) and (X^*, f^*) are dual filtered complexes.*

Proof. Let $\sigma_0, \sigma_1, \ldots, \sigma_n$ be a linear ordering of cells compatible with (X, f) and $\sigma_n^*, \sigma_{n-1}^*, \ldots, \sigma_0^*$ be the corresponding ordering of the dual X^*. Note that σ_i^* is the $(n-i)$-th cell in the dual ordering. It follows that:

$$\sigma_i \preceq \sigma_j \text{ implies } i < j \Leftrightarrow \sigma_j^* \preceq \sigma_i^* \text{ implies } n - j < n - i$$

and

$$i < j \text{ implies } f(\sigma_i) \leq f(\sigma_j) \Leftrightarrow n - j < n - i \text{ implies } f^*(\sigma_j^*) \leq f^*(\sigma_i^*).$$

This shows that the linear ordering on X is compatible with f if and only if the dual linear ordering on X^* is compatible with f^*, as required. □

2.4.2 Categorical Definition

We recall here a general categorical definition of persistent homology. We mention this definition for the sake of completeness, but it is not necessary for the understanding of this thesis. We will use the more algebraic notions of this section only in Chapter 7. We cover a more combinatorial definition of persistent homology in the next section.

Given a filtered complex (X, f), there are inclusions $f^{-1}(-\infty, r] \to f^{-1}(-\infty, s]$ of sublevel sets for $r \leq s$. Applying degree-k homology with coefficients in a field $\mathbb{K}$ to these inclusions yields linear maps between vector spaces

$$H_k(f^{-1}(-\infty, r]) \to H_k(f^{-1}(-\infty, s]).$$

The resulting functor $H_k(f) : (\mathbb{R}, \leq) \to \mathsf{Vec}_{\mathbb{K}}$ from the poset category $(\mathbb{R}, \leq)$ to the category of vector spaces over the field $\mathbb{K}$ is called a persistence module. More generally, a persistence module needs not come from a filtration function.

Definition 2.29. A *persistence module* is a functor

$$F : (\mathbb{R}, \leq) \to \mathsf{Vec}_{\mathbb{K}}$$

where $(\mathbb{R}, \leq)$ is the real line with its total ordering $\leq$. A persistence module is *pointwise finite-dimensional* if all the vector spaces $F(t)$ are finite-dimensional. An *interval module* is a persistence module $\mathbb{K}_I$ that is rank 1 on an interval $I \subseteq \mathbb{R}$ with identity maps internal to I and 0 elsewhere.

In this thesis, all persistent modules come from the sublevel sets of a function $f : X \longrightarrow \mathbb{R}$, that is, $F = H_k(f)$. By [29] we have the following (fundamental) decomposition theorem for persistence modules.

Theorem 2.30. *(Crawley-Boevey [29]) Any pointwise finite-dimensional persistence module F is isomorphic to a direct sum of interval modules*

$$F \cong \bigoplus_{l \in L} \mathbb{I}_{[b_l, d_l)},$$

and this decomposition is unique up to reordering.

Each interval summand $\mathbb{I}_{[b_l, d_l)}$ represents a degree-k homological feature that is *born* at $r = b_l$ and *dies* at $r = d_l$. If the final space X has non-trivial homology there are features that never die, then, these have $d_l = \infty$, and the interval is called *essential.*

Typically, tame functions $f : X \longrightarrow \mathbb{R}$ have finitely many critical values $\alpha_0, \cdots, \alpha_n \in \mathbb{R}$ where the homology changes: the sublevel sets X_{t_1}, X_{t_2} have the same homology when $t_1, t_2 \in (\alpha_i, \alpha_{i+1})$ for $i = 0, \cdots, n-1$. Therefore, they fall into the framework of Theorem 2.30.

There are several ways to represent these interval modules. The most commonly used is the *degree-k persistence diagram* of f, which is the multiset

$$\mathsf{Dgm}^k(f) = \{\, [b_l, d_l) \mid l \in L \,\},$$

where $H_k(f) \cong \bigoplus_{l \in L} \mathbb{I}_{[b_l,d_l)}$. We write $[b_l, d_l)_k \in \mathsf{Dgm}^k(f)$ to denote the homological degree of an interval and define the *persistence diagram* of f as the disjoint union over all degrees:

$$\mathsf{Dgm}(f) = \bigsqcup_{k=0}^{\dim(X)} \mathsf{Dgm}^k(f).$$

Writing $\mathsf{Dgm}_{\mathbf{F}}(f)$ for the multiset of finite intervals with $d_l < \infty$, and $\mathsf{Dgm}_\infty(f)$ for the remaining essential ones, we obtain $\mathsf{Dgm}(f) = \mathsf{Dgm}_{\mathbf{F}}(f) \sqcup \mathsf{Dgm}_\infty(f)$.

Later, in Section 3.2 we introduce the notion of barcodes, which is an equivalent way to represent pointwise finite-dimensional persistence modules.

2.4.3 Computations

In this section, we describe briefly how persistence diagrams are computed in practice for a filtered cell complex (X, f). We restrict ourselves to the case $\mathbb{K} = \mathbb{F}_2$, the field of two elements. To compute the persistence diagram $\mathsf{Dgm}(f)$, we choose an ordering $\sigma_0, \sigma_1, \dots, \sigma_n$ of the cells in X that is compatible with f. Cells σ_i and σ_j appear at the same step in the nested sequence of sublevel sets $\left(f^{-1}(\infty, r]\right)_{r \in \mathbb{R}}$ if $f(\sigma_i) = f(\sigma_j)$. For the following computations however, we must add exactly one cell at every step:

$$\emptyset \subset \{\, \sigma_0 \,\} \subset \{\, \sigma_0, \sigma_1 \,\} \subset \cdots \subset \{\, \sigma_0, \sigma_1, \dots, \sigma_{n-1} \,\} \subset \{\, \sigma_0, \sigma_1, \dots, \sigma_n \,\} = X.$$

When adding the cells one step at a time, a cell of dimension k causes either the birth of a k-dimensional feature or the death of a $(k-1)$-homology class [38], that is, each cell is either a *birth* or a *death cell.* A pair (σ_i, σ_j) of cells where σ_j kills the homological feature created by σ_i is called a *persistence pair.* A persistence pair (σ_i, σ_j) corresponds to the interval $[f(\sigma_i), f(\sigma_j)) \in \mathsf{Dgm}_{\mathbf{F}}(f)$. Note that this interval can be empty, namely if $f(\sigma_i) = f(\sigma_j)$. Empty intervals are usually neglected in the persistence diagram. A birth cell σ_i with no corresponding death cell is called *essential,* and corresponds to the interval $[f(\sigma_i), \infty) \in \mathsf{Dgm}_\infty(f)$.

Recall that presentations for the standard homology groups are found by studying the image and kernel of integer-entry matrices that represent the boundary maps taking oriented chains of dimension k to those of dimension $(k-1)$ [86]. In persistent homology, we work with the $\mathbb{F}_2$ *total boundary matrix* D, which is defined by $D_{i,j} = 1$ if $\sigma_i \lhd \sigma_j$ and 0 otherwise. Define

$$r_D(i,j) = \operatorname{rank} D_i^j - \operatorname{rank} D_i^{j-1} - \operatorname{rank} D_{i+1}^j + \operatorname{rank} D_{i+1}^{j-1}$$

where $D_i^j = D[i:n, 0:j]$ is the lower-left sub-matrix of D attained by deleting the first rows up to $i-1$ and the last columns starting from $j+1$.

Theorem 2.31 (Pairing Uniqueness Lemma [28])**.** *Given a linear ordering of the cells in a filtered cell complex X, (σ_i, σ_j) is a persistence pair if and only if $r_D(i, j) = 1$.*

The ranks are usually computed by applying the column reduction algorithm [45] to obtain the reduced matrix R and using the property that $\operatorname{rank} D_i^j = \operatorname{rank} R_i^j$ under the operations of the algorithm. The persistence pairs can then be read off easily since $r_R(i, j) = 1$ if and only if the ith entry of the jth column of the reduced matrix is the lowest 1 of this column. However, here, we can work directly with r_D.

Corollary 2.32. *If $r_D(i, j) \neq 1$ and $r_D(j, i) \neq 1$ for all j then the cell σ_i is essential.*

Proof. The fact that every cell is either a birth or a death cell implies that σ_i must be an unpaired birth or death cell. However, as every filtration begins as the empty set, there are no unpaired death cells. □

Chapter 3

Trees and Barcodes

3.1 Trees

We finally introduce the main topic of study of this thesis: trees. There are many notions of trees in mathematics. From modeling species genealogy with phylogenetic trees, modeling roots, neurons or rivers with trees and studying the sublevel sets of a function using its merge tree, the definition of tree differs a bit depending on the context or application. Here, we explore several mathematical definitions of trees and explain their differences. The definitions in this section have been formalized in [32], together with J. Curry, J. DeSha, K. Hess, L. Kanari and B. Mallery. All these notions are based on combinatorial trees, which we introduce now.

3.1.1 Combinatorial Trees

We start with the simplest definition, that of a combinatorial tree.

Definition 3.1. A *combinatorial tree* T is a connected, acyclic, binary graph. It is *finite* if the number of vertices is finite. A *rooted tree* is a combinatorial tree with a distinguished vertex of degree 1 called the *root.* Non-root vertices of degree 1 are called *leaves*, and vertices of degree 3 are called *bifurcations* or *internal nodes.* We assume that there are no vertices of degree 2. A combinatorial tree equipped with an embedding into $\mathbb{R}^3$ is called a *geometric tree.* They inherit naturally the topology of $\mathbb{R}^3$, forming the set of geometric trees that we denote by $\mathcal{T}$.

A *labelling* of a combinatorial tree T is a bijective map from its set of vertices $V(T)$ to a set S of labels. A labelling is *ordered* if S is a subset of of the natural numbers $\mathbb{N}$. An ordered labelling of a tree with n vertices gives rise to an $n \times n$ adjacency matrix, of which the (i, j)-coefficient is 1 if there is an edge between the vertices labelled i and j and is 0 otherwise.

Two combinatorial trees T and T' are *isomorphic* if they are isomorphic as graphs. Equivalently, T and T' are isomorphic if there exist ordered labellings of both with respect to which their adjacency matrices are identical.

When rooted trees are considered, there is a natural way to induce an orientation on the edges of the tree: for each vertex v, there is a unique path from v to the root r. Every edge of the tree is oriented from the vertex further from r to the closer one (with respect to the graph-path distance).

A vertex v of T is a *parent* of a vertex w if there is a directed edge from w to v; the vertex w is then a *child* of v. If there is a sequence of directed edges from w to v, then v is an *ancestor* of w. The least common ancestor between two vertices w and w' is the first (in terms of graph-path distance) vertex v that is an ancestor of both, see Figure 3.1. Each vertex of T has a unique parent, except for the root r, which has no parent at all.

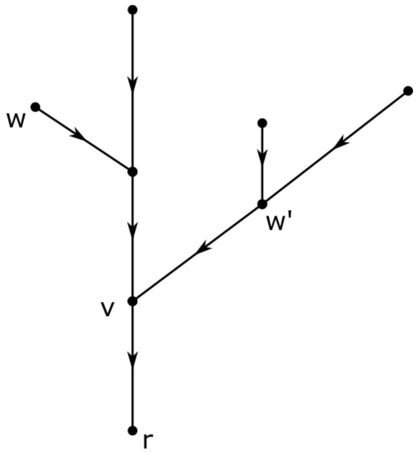


Figure 3.1: A combinatorial tree T seen as a directed graph with its root r. The least common ancestor of w and w' is the vertex v. The path-distance between w and v is 2, as there are two edges on the unique directed path between them.

Remark 3.2. The graph-path distance between two nodes of a graph, sometimes also called the *hop-metric*, is the number of edges on the (unique) shortest path between the two nodes. The notion of v being an ancestor of w is equivalent to v being on the path between w and the root r, hence the distance between v and r in the path distance is shorter than the one between w and r.

Note that a finite combinatorial tree T is fully specified by its set of vertices, equipped with the partial order specified by the "is a parent of" relation. This turns the set of vertices of a tree $V(T)$ into a poset, which has T as a Hasse diagram.

3.1.2 Merge Trees and Combinatorial Merge Trees

The language of "parents" and "children" obviously comes from studying ancestral relations for people (family trees) and species (phylogenetic trees). There are also situations where the parent-child relation is determined in part by a notion of "height," which is how merge trees are defined.

Definition 3.3. A *merge tree* is a rooted combinatorial tree T, together with a function on the vertices $h : V(T) \longrightarrow \mathbb{R} \cup \{\infty\}$, called a *height function*, that satisfies two properties.

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

A *generic merge tree* is a merge tree (T, h) such that the height function $h : V(T) \to \mathbb{R}$ is injective. We always assume our merge trees are generic, unless otherwise indicated.

Two merge trees (T, h) and (T', h') are *isomorphic* if there is a graph isomorphism $\varphi : T \to T'$ that preserves heights, i.e., $h = h' \circ \varphi$. In this thesis, we will not make a distinction between merge trees and isomorphism-equivalence classes of merge trees.

Remark 3.4 (Drawing Conventions for Merge Trees). Many authors choose to draw merge trees so that the function $h : V(T) \to \mathbb{R}$ resembles height when embedded in the page. This has the effect of placing the root node higher than the leaf nodes, contrary to how trees appear in nature. To honor the natural orientation and size of trees in nature, we draw our merge trees with the opposite convention, so that the root is lower than the leaves and so that $f(r) = \infty$ is represented with a finite value.

Remark 3.5 (Alternative Definition of Merge Trees). Another, perhaps more common, definition of a merge tree is that it is the Reeb graph of the epigraph of a function. From this point of view, the merge tree T of a real-valued function $f : X \to \mathbb{R}$ is the quotient space of the epigraph $\Gamma^+ := \{(x, t) \in X \times \mathbb{R} \mid f(x) \leq t\}$ by the equivalence relation specified by $(x, t) \sim (y, s)$ if and only if $s = t$ and x and y are in the same path component of the sublevel set filtration of f at t, i.e., $[x] = [y] \in \pi_0(f^{-1}(-\infty, t])$. Since the projection map from Γ^+ onto the second coordinate is constant on equivalence classes, this projection map factors to define the height function. Under reasonable tameness conditions, the quotient space is homeomorphic to the geometric realization of a combinatorial tree, where vertices correspond to connected components of "critical" points.

Example 3.6. A typical example of merge tree is one arising from measuring height on an embedded manifold $X \subseteq \mathbb{R}^n$. Here "height" can be thought of as the scalar product with a specified unit vector, or as a Morse function. Figure 3.2 shows a simple example of a topological space and the corresponding merge tree.

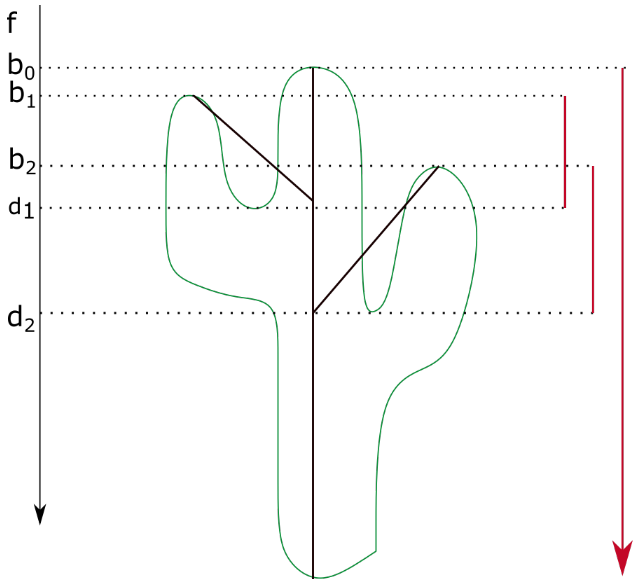


Figure 3.2: A circle X is embedded in $\mathbb{R}^2$ and drawn in green to resemble a cactus with the height function f measuring distance down the page. The corresponding merge tree (Definition 3.3) is drawn in black. The barcode of the persistence module in degree 0 (Definition 2.29) associated to (X, f) is shown in red on the right.

Merge trees defined as quotient of topological spaces can be equipped with a distance called the interleaving distance [85]. The definition as stated in [85] does not apply to the setting of Definition 3.3 because it requires the height function h to be defined on the whole topological space T and not only on the vertices. For the following definition, we require that T be considered as a embedded tree in $\mathbb{R}^3$ (a geometric tree) and that the height function $h : T \longrightarrow \mathbb{R}$ is defined on any point of T, not only on the vertices. We denote such an embedding by $|T|$. It is easy to see how a height function can be extended to all the points of an edge: it suffices to take the interpolation of the function between the two end-vertices of the edge.

Definition 3.7. [59] Let $(T, h), (T', h')$ be two merge trees with height functions h, h'. A *ε-good map* $\alpha : (T, h) \longrightarrow (T', h')$ is a continuous map such that the following conditions hold.

1. For every $x \in |T|$, $|h'(\alpha(x)) - h(x)| \leq \varepsilon$.
2. For $v \in \mathrm{Im}(\alpha)$, let x' be the least common ancestor of $\alpha^{-1}(v) \in T$. Then $|h(x') - h(w)| \leq 2\varepsilon$ for all $w \in \alpha^{-1}(v)$.

3. For all $v \neq \mathrm{Im}(\alpha)$, $\mathrm{depth}(v) \leq \varepsilon$, where $\mathrm{depth}(v)$ is the largest height difference between v and a node in the subtree rooted at v.

The *interleaving distance* between two merge trees T and T' is

$$d_I\big((T,h),(T',h')\big) = \inf\{\varepsilon \mid \text{there is an } \varepsilon\text{-good map between } T \text{ and } T'\}.$$

Example 3.8. Figure 3.3 shows an example of two merge trees at distance ε.

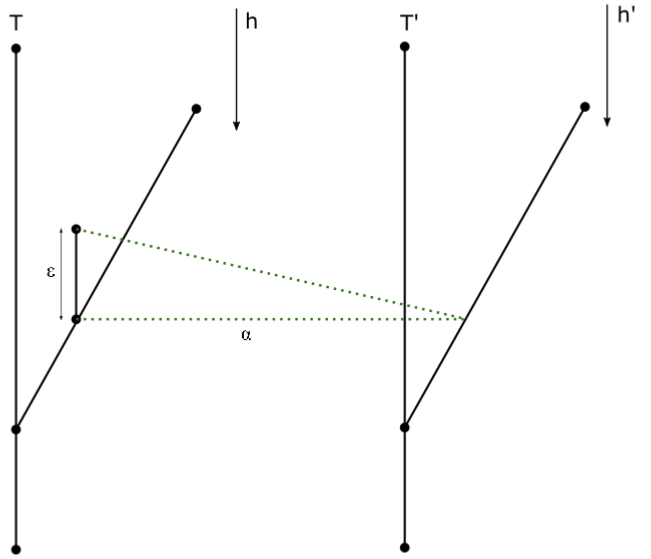


Figure 3.3: Two trees at interleaving distance ε. The ε-good map α depicted in green dotted lines sends each point of T to the corresponding point in T', except the small branch of size ε which is sent to the same point in T'.

There is a natural ordered labelling on the vertices of a generic merge tree (T,h), inherited from the function h, by ordering the vertices according to their h-value: the leaf node with lowest h-value is labelled 0, and the remaining nodes are labelled based thereafter on the order in which they appear. We call the labels on the leaves the *birth labels* and the ones on the internal vertices the *death labels*, for reasons that will become clear later when we study the relationship between trees and barcodes.

We are now in a position to state the first novel definition of this section. Recall that two graphs are isomorphic if they admit ordered labellings with respect to which their adjacency matrices are the same. A merge tree includes the additional data of heights of each node. By focusing separately on the order of births and the order of deaths, along with adjacency data, we have a more flexible notion of equivalence between merge trees.

Definition 3.9. Two generic merge trees (T, h) and (T', h') are *combinatorially equivalent* if they are isomorphic as graphs via a graph isomorphism preserving the orders of births and of deaths, respectively. In more detail, (T, h) and (T', h') are combinatorially equivalent if there exists a graph isomorphism $\varphi : T \to T'$ such that the following conditions hold.

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

We note that these two conditions specify two different sets for the logical quantifier and that the total order on vertices need not be preserved; see Figure 3.4 for an example.

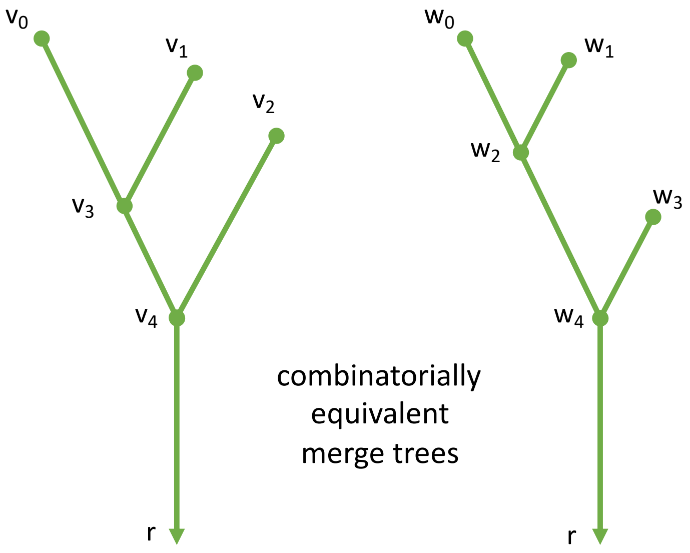


Figure 3.4: Two combinatorially equivalent merge trees are shown. Notice that the total order of the vertices is not preserved, but the orders among leaf nodes and internal nodes are preserved separately.

Remark 3.10 (Combinatorial Merge Trees)*.* Note that that combinatorial equivalence classes of merge trees are simply combinatorial trees equipped with an ordered labelling L_l of the leaves (a birth label) and an ordered labelling L_i of the internal nodes (a death label) such that the label $L_i(v)$ of internal node v is larger than the label $L_i(w)$ of internal node w if v is an ancestor of w. We call such a tree a *combinatorial merge tree.*

Example 3.11 (Translation Invariance)**.** Consider two generic merge trees (T, h) and (T, h') such that $h' = h + \Delta$ for some real number Δ. We say (T', h') is a *translation* of T. A generic merge tree is combinatorially equivalent to any translation of itself. However, combinatorial equivalence detects relationships more general than translation; see Figure 3.4.

3.1.3 Phylogenetic Trees and Combinatorial Phylogenetic Trees

As mentioned earlier, most of the language concerning trees is inspired by the study of ancestral relationships. Although trees have been used for this purpose for centuries, a formal definition of a phylogenetic tree—and more importantly a clear coordinatization on the set of all phylogenetic trees—was given only somewhat recently in the landmark paper of Billera, Holmes and Vogtmann [13]. We review some of these definitions, modifying the terminology slightly for our purposes.

Definition 3.12. A *(metric) phylogenetic tree* is a rooted combinatorial tree T endowed with

1. a labelling of the leaf nodes, and
2. a non-negative real number associated to every parent-child pair.

The values assigned to each parent-child pair can be considered as weights on the graph edges. By contrast, a *combinatorial phylogenetic tree* is a rooted combinatorial tree with just a labelling of the leaf nodes.

Example 3.13. Figure 5.5B shows all combinatorial classes of merge trees with four leaves and Figure 5.5C shows all combinatorial classes of phylogenetic trees with four leaves.

3.1.4 Different Spaces of Trees

In this section, we describe the difference between phylogenetic trees and merge trees, which consists of a minor distinction about the labelling that creates impactful differences between the two spaces. This is based on joint work with J. Curry, J. DeSha, K. Hess, L. Kanari, and B. Mallery [32].

One of the key differences between metric phylogenetic trees and merge trees is that phylogenetic trees always have labelled leaf nodes, with labels independent of the lengths on the edges. This makes sense because the BHV space—the set of all possible metric phylogenetic trees on n leaf nodes, denoted $\mathcal{MPT}_n$—documents all possible evolutionary relationships among n fixed species. The labels matter because the involved species matter.

We denote the set of all merge trees with n leaf nodes (Definition 3.3) by $\mathcal{MT}_n$. We consider also the set $\mathcal{LMT}_n$ of labelled merge trees with n leaves, where the labelling is arbitrary (see Definition 3.1). Let

$$\mathcal{I} : \mathcal{LMT}_n \longrightarrow \mathcal{MT}_n,$$

denote the map that sends a labelled merge tree to its corresponding unlabelled merge tree, forgetting the labels.

We describe the relationship between these two types of tree spaces in the following proposition.

Proposition 3.14. *For every $\Delta \in \mathbb{R}$, there is an injective map from the set of metric phylogenetic trees with n leaves, $\mathcal{MPT}_n$, to the set of labelled merge trees with n leaves, $\mathcal{LMT}_n$*

$$\mathcal{H}_\Delta : \mathcal{MPT}_n \longrightarrow \mathcal{LMT}_n.$$

such that the composite $\mathcal{I} \circ \mathcal{H}_\Delta$ has a fiber of cardinality $n!$ over generic merge trees, corresponding to permutations of the labels on the leaf nodes. Moreover, if $\Delta \geq 0$, there is a natural section of $\mathcal{I} \circ H_\Delta$,

$$\mathcal{T}_\Delta : \mathcal{MT}_{n,generic} \longrightarrow \mathcal{MPT}_n,$$

that sends a generic merge tree to a metric phylogenetic tree that is labelled by birth order and where the distance from the root node to its child is Δ.

Proof. Given a metric phylogenetic structure on a rooted tree T, we can define a height function h on T as follows. Every node v that is not the root node r is assigned the function value $h(v) := \Delta - d(r, v)$, where d is the sum of the weights of each edge along the unique path connecting r to v. This defines the map $\mathcal{H}_\Delta$ in the statement of the proposition.

As explained earlier, every generic merge tree admits a canonical ordering of its leaf nodes by height order. If two generic labelled merge trees in the image of $\mathcal{H}_\Delta$ are isomorphic as merge trees, then there is a unique permutation of the n leaf labels taking one labelling to the other. This proves the second statement.

Finally, the map $\mathcal{T}_\Delta$ sends a generic unlabelled merge tree (T, h) to the metric phylogenetic structure on T that has labels given by birth order and where the weight on an edge is given by the difference in heights of its two vertices. The distance from the root node to its child is given by Δ. □

Remark 3.15. Each of the three sets above can be equipped with topologies. In [13], the space of phylogenetic trees is topologized as a CAT(0) space where each orthant records a distinct *split topology* [13]. Both labelled merge trees and merge trees can be topologized using versions of the interleaving distance [85], Definition 3.7. Unfortunately, the map $\mathcal{T}_\Delta$ is discontinuous with respect to these topologies, as can be seen from Figure 3.5.

Example 3.16 (Sensitivity to Generators). Although the two merge trees in Figure 3.5 are isomorphic as graphs, the only possible graph isomorphism reverses the birth order, hence these generic merge trees are not combinatorially equivalent. When considering the persistent homology of the height filtration for these two trees (see Section 2.4), the homology generator of the essential class starts with the node labelled by 0 or A on the left hand side, while on the right hand side, it starts with the 0 or B label. This is sometimes called "instability" or "sensitivity" of generators in TDA. Together with Figure 3.4, these specify the three possible combinatorial equivalence classes of merge trees with three leaf nodes.

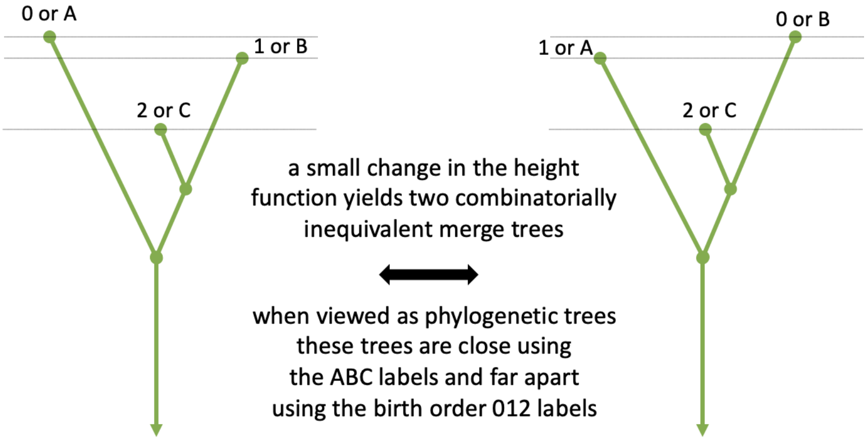

0 or A
1 or B
2 or C
1 or A
0 or B
2 or C
a small change in the height
function yields two combinatorially
inequivalent merge trees
when viewed as phylogenetic trees
these trees are close using
the ABC labels and far apart
using the birth order 012 labels

Figure 3.5: Two generic merge trees that are isomorphic as graphs. When they are regarded as phylogenetic trees, we fix alphabetical ('ABC') names for the leaf nodes, as if the nodes represented species that went extinct at different times. With this labelling they are considered close in the phylogenetic tree metric defined by [13]. When they are regarded as merge trees, they are unlabelled and are close in the interleaving distance [85]. However, if we use the birth order ('012') to label the leaf nodes and regard them as phylogenetic trees, then they are far apart in the phylogenetic tree metric. This shows the discontinuity of the map $\mathcal{T}_\Delta$ of Theorem 3.14.

Proposition 3.14 shows that, despite their apparent similarity, there are significant differences between metric phylogenetic trees and merge trees. Indeed neither of the maps above is a bijection. However, if one quotients the set of labelled merge trees by translations of the height function then the map induced by $\mathcal{H}_\Delta$ should be a bijection; alternatively one could modify the definition of merge trees so that the root node has a fixed height N, as in the drawing convention of Remark 3.4.

Although the proposition and remark above identify certain differences and similarities between metric phylogenetic trees and merge trees, for this thesis the most important distinction is in terms of combinatorial type. In this respect, merge trees and phylogenetic trees are distinguished by the explicit ordering of birth and death nodes. This observation will lead to different formulas for the numbers of top-dimensional strata in the set of combinatorial phylogenetic trees $\mathcal{PT}_n$, which is $(2n-1)!!$, and in $\mathcal{MT}_n$, which is $(n+1)!n!2^{-n}$. For now, however, the reader is encouraged to consult Table 3.1 and Figure 3.6 for two convenient summaries of the similarities and differences between combinatorial trees, merge trees, (combinatorial) phylogenetic trees, and barcodes.

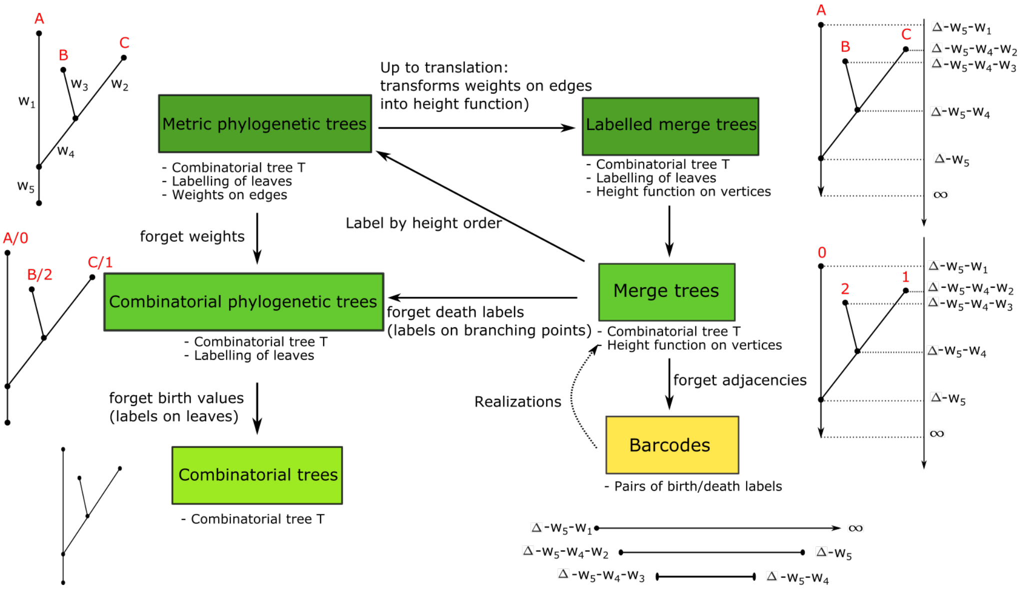


Figure 3.6: Summary of the different notions of tree studied in this thesis and their relations, as expressed in part by Proposition 3.14. One can turn a metric phylogenetic tree (with labels A,B,C in red) into a labelled merge tree. Generic merge trees can be turned into metric phylogenetic trees by labelling according to birth order (labels $0, 1, 2$ in red), but this process is not continuous.

3.2 Barcodes

In Section 2.4, we introduced the notion of persistent homology and persistence diagrams from an algebraic point of view. Here, we consider a notion equivalent to that of a persistence diagram but with a more combinatorial approach. For most of the remainder of this thesis, apart from Chapter 7, we will work with barcodes instead of persistence diagrams.

3.2.1 Barcodes and Combinatorial Barcodes

Let (X, f) be a filtered space and $H_k(f)$ its associated persistence module with decomposition $H_k(f) \cong \bigoplus_{j \in \mathcal{J}} \mathbb{K}_{I_j}^{\oplus n_j}$. Recall that the k-th persistence diagram, or barcode, of f is the multiset

$$\mathsf{Dgm}^k(f) = \{I_j\}_{j \in \mathcal{J}}.$$

In most applications, each interval I_j is of the form $[b_j, d_j)$, where b_j is the birth of the homological feature corresponding to I_j and d_j its *death.* We call the interval $[b_j, d_j)$ a *bar* in the barcode B. Because this thesis is mainly about combinatorial aspects of barcodes, we consider only the pairing of the births and deaths, leading us to work instead with pairs of points instead of intervals.

Definition 3.17. A *barcode* $\{(b_i, d_i)\}_{i \in J}$ is a multiset of pairs such that $-\infty < b_i < d_i < \infty$ for each $i \in J$. The first coordinate b_i is called the *birth* and the second one d_i is called the *death.* If there is a bar (b_0, d_0) that contains all the others, it is called *essential.* We denote the set of barcodes with n bars by $\mathcal{B}_n$. In Chapters 3, 5 and 6, $\mathcal{B}_n$ consists of the set of barcodes with $n + 1$ bars, because we also count the essential bar.

In this thesis, we represent barcodes graphically by drawing the interval between b_j and d_j for each index j.

Remark 3.18. The notations for barcodes slightly differ in the chapters of this thesis depending on the context. When studying the set of barcodes, we do not need the essential bar (b_0, d_0). When considering barcodes computed from trees, such a bar always exists. We recall the main notation at the beginning of each chapter.

Remark 3.19. Notice that in the new definition, we drop the interval notation to define a barcode as a pair of birth and death values. The reader familiar with the algebraic setting of persistent homology will also notice that we suppose that the bars corresponding to essential classes have finite values instead of being half-open intervals. In practice, such essential classes are given finite values to be stored in the computer. Sometimes, we denote $d_0 = \infty$ as well, but the reader should keep in mind that we are mostly interested in the combinatorial aspects of the pairs (b_i, d_i).

Example 3.20. Figure 3.7 shows an example of a barcode with two different indexings.

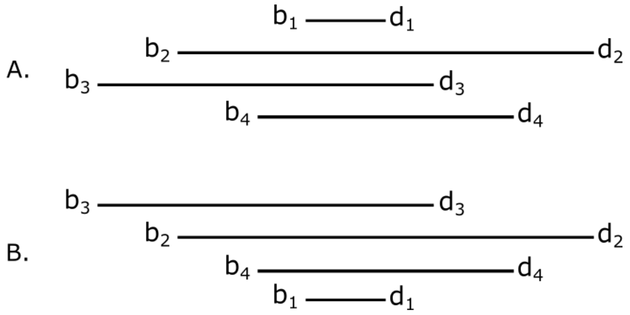


Figure 3.7: (A) A barcode with 4 bars. (B) The exact same barcode with a different indexing where the bars are ordered by increasing birth times.

Although in general a barcode can be a true multiset, in this thesis we are concerned primarily with barcodes that are actually sets, leading us to formulate the following definition. In practice, the indexing set J is commonly the set $\{1, ..., n\}$. This gives the bars in the barcode an arbitrary but fixed indexing. It can sometimes be convenient to assume that the indexing is such that the births are ordered increasingly $b_1 < b_2 < ... < b_n$, and we will specify when this is the case.

Definition 3.21. A barcode B is *strict* if $b_i \neq b_j$, $d_i \neq d_j$ if $i \neq j$. We denote the set of strict barcodes with n bars by $\mathcal{B}_n^{st}$.

As for merge trees (Remark 3.10), we can combinatorially reduce the information of a barcode by computing the ordering of the deaths with respect to the ordering of the births. This was first observed in [69].

Definition 3.22. Let $B = \{(b_i, d_i)\}_{i \in \{1,...,n\}}$ be a strict barcode. If we order the births increasingly such that $b_{i_1} < \ldots < b_{i_n}$, the indexing in $\{1, ..., n\}$ gives a permutation τ_b given by $\tau_b(k) = i_k$, so that

$$b_{\tau_b(1)} < \ldots < b_{\tau_b(n)}. \tag{3.1}$$

Similarly, ordering the deaths $d_{j_1} < \ldots < d_{j_n}$ gives rise to a permutation τ_d with $\tau_d(k) = j_k$. The *permutation* σ_B *associated to* B is defined as $\sigma_B = \tau_b^{-1}\tau_d$; it tracks the ordering of the death values with respect to the birth values. Two strict barcodes are *combinatorially equivalent* if they have the same associated permutation.

Though it consists only of a permutation in Sym_n, σ_B will sometimes be called a *combinatorial barcode*, for reasons that will become clear in and Section 3.3.3 and Figure 3.10.

Remark 3.23. The permutations τ_b and τ_d both depend on the indexing choice of the b_i and d_i. However, the permutation σ does not depend on any indexing of the births and deaths, it is intrinsic to the multiset B. Indeed, σ_B can be defined directly as the permutation that sends the i-th death (in increasing order) to the $\sigma(i)$-th birth (idem). If we assume that the births are ordered increasingly, then $\tau_b = \mathrm{id}$ and σ_B can be defined directly by $\sigma_B = [j_1 j_2 \dots j_n]$, the indices of the deaths when they are ordered increasingly.

Example 3.24. Figure 3.7A shows an example of a strict barcode. Its birth permutation is $\tau_b = [3241]$, since

$$b_3 < b_2 < b_4 < b_1.$$

Similarly, its death permutation is $\tau_d = [1342]$, since $d_1 < d_3 < d_4 < d_2$. The permutation σ_B associated to the barcode of Figure 3.7A is $\sigma_B = [4132] = \tau_b^{-1}\tau_d$. Figure 3.7B shows the same barcode with the bars ordered by birth times. The corresponding permutations $\tau_b = [1234]$ and $\tau_d = [4132]$ are different, but the product $\sigma_B = \tau_b^{-1}\tau_d = [4132]$ is the same, as it does not depend on the indexing of the bars. Further examples are depicted in Figure 3.8.

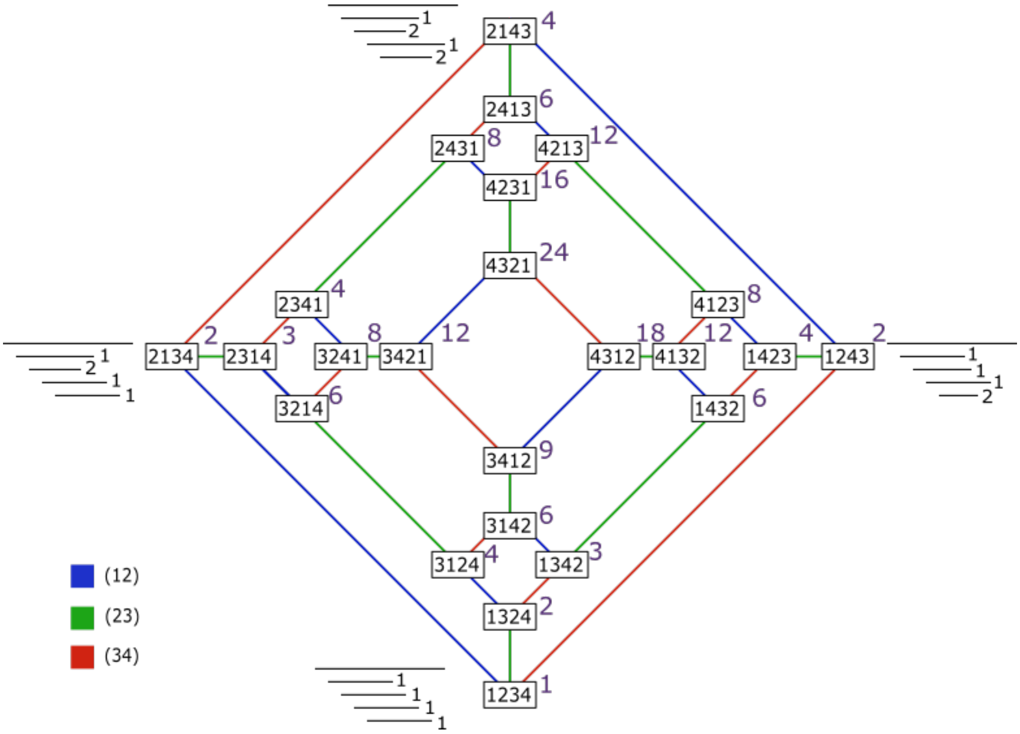


Figure 3.8: The Cayley graph of Sym_4 generated by the three transpositions $(12), (23), (34)$. Four barcodes are drawn next to the extremities of the graphs (permutations $[1234], [2134], [2143], [1243]$) to illustrate a typical barcode corresponding to each permutation. The value next to each permutation is the tree realization number, introduced in Section 5.1.

We extend Theorem 3.22 to non-strict barcodes in Section 4.2.3.

3.2.2 Distances for Barcodes

To turn the set $\mathcal{B}_n$ of barcodes with n bars into a topological space, one needs a topology. One way to do this is by introducing the bottleneck or Wasserstein distances, two commonly used metrics for barcodes. Intuitively, the bottleneck distance between two barcodes B and B' tries all possible matchings between the bars of B and the bars of B' and chooses the one that minimises the “energy” required to move the matched pair of bars with maximal separation. However, it does not consider only matching of bars between B and B' but also with points on the diagonal $\Delta = \{(x, x) \mid x \in \mathbb{R}\}$.

Definition 3.25. Let $B = \{(b_i, d_i)\}_{i\in\{1,\dots,n\}}$ and $B' = \{(b'_i, d'_i)\}_{i\in\{1,\dots,m\}}$ be two barcodes. The *bottleneck distance* between B and B' is

$$d_B(B, B') = \min_{\gamma} \max_{x\in B} \|x - \gamma(x)\|_\infty,$$

where γ runs over all possible matchings, i.e., maps that assign to each bar $(b_i, d_i) \in B$ either a bar in B' or a point in the diagonal Δ, such that no point of B' is in the image more than once. Here, $\|\cdot\|_\infty$ is the l^∞-norm on $\mathbb{R}^2$.

Remark 3.26. The permutation γ acts as a “reindexing” of the indices of B and B', and in particular ensures that $d_B(B, B')$ does not depend on any indexing of the bars.

The *Wasserstein distance* is defined in a similar way by taking the sum over all l_2-distances between x and $\gamma(x)$ instead:

$$d_W(B, B') = \min_{\gamma} \sqrt{\Sigma_{x\in B} \|x - \gamma(x)\|_2^2)}.$$

Remark 3.27. Note that in general, the barcodes B and B' need not have the same number of bars. The diagonal allows matchings between barcodes with different number of bars, since “ummatched” bars can be sent to the diagonal. In this thesis however, we are study the set of barcodes $\mathcal{B}_n$ with exactly n bars (for arbitrary, but fixed n) and restrict ourselves to this case.

We are mainly interested in $\mathcal{B}_n$ as a set and the main results we prove do not depend on the metric that is chosen on $\mathcal{B}_n$. We will still with a slight abuse of notation mostly talk of $\mathcal{B}_n$ as a *space*, without specifying a specific metric on it. An exception to that is Section 4.2.4, where we explain how a metric $\tilde{d}_B$ on $\mathcal{B}_n$, which is closely related to the bottleneck distance, occurs in an alternative description of the *set* $\mathcal{B}_n$ that we work with later on.

An important result in TDA is the stability of persistence homology.

Theorem 3.28 (Stability of persistence diagrams)**.** *[26, 102] Let $f, g : X \longrightarrow \mathbb{R}$ be two tame functions on X, a triangulable, compact metric space, and let B, B' be the barcodes of their respective sublevel sets. Then,*

$$d_B(B, B') \leq \|f - g\|_\infty.$$

3.2.3 Statistics on Barcodes

Unfortunately, the space of barcodes equipped with the bottleneck or Wasserstein metric is not a Hilbert space, making it difficult to do statistics. A lot of work has been done in trying to build *kernels* to map the space of barcodes into a Hilbert space, which enables us to turn non-linear methods on a space $\mathcal{X}$ into linear ones in a feature space, a Hilbert space $\mathcal{H}$.

More precisely, a *kernel* on X is a map

$$k : \mathcal{X} \times \mathcal{X} \longrightarrow \mathbb{R} :$$

$$k(x, y) = \langle \phi(x), \phi(y) \rangle_{\mathcal{H}},$$

where $\phi : \mathcal{X} \longrightarrow \mathcal{H}$ is a well-chosen map and $\langle \cdot, \cdot \rangle_{\mathcal{H}}$ is the scalar product in $\mathcal{H}$. Using kernels allows us to study statistics on $\mathcal{X}$ in the feature space $\mathcal{H}$, which has a scalar product.

Kernel methods in TDA include persistence images [2] and persistence landscapes [21] for example.

In this thesis, we introduce new methods to study statistics on the space of barcodes based on the field of permutation statistics, see Chapter 4. Here, we introduce basic statistics on barcodes that will be used later on.

Definition 3.29. Let $B = \{(b_i, d_i)\}_{i \in \{1,\dots,n\}}$ be a barcode with n bars. The *average of the birth times* is

$$\bar{b} = \frac{1}{n} \sum_i b_i$$

and the *average of the death times* is

$$\bar{d} = \frac{1}{n} \sum_i d_i.$$

The *standard deviation of the birth times* is

$$\|v_b\| = \left(\sum_{i=1}^{n} |b_i - \bar{b}|^2 \right)^{1/2}$$

and the *standard deviation of the death times* is

$$\|v_d\| = \left(\sum_{i=1}^{n} |d_i - \bar{d}|^2 \right)^{1/2}.$$

The *persistent entropy* [8] of B is defined as

$$E(B) = -\sum_{i=1}^{n} \frac{l_i}{L} \log\left(\frac{l_i}{L}\right),$$

where $l_i = d_i - b_i$ is the length, or persistence, of the interval corresponding to (b_i, d_i), and $L = \sum_{i=1}^{n} l_i$.

The last statistic that will be used in this thesis is the *tree-realization number* that we define in Theorem 5.1.

3.3 From Trees to Barcodes

In Section 2.4 we recalled the persistent homology of a filtration. In this section, we restrict ourselves to the specific case of merge trees. As explained in Remark 3.5, a merge tree can be defined as the Reeb graph of the epigraph of a function. By thinking of a merge tree $|T|$ embedded in $\mathbb{R}^n$ and considering the sublevel sets of the height function h, one can compute its barcode as in Section 2.4. It can also be done directly from the tree considered as a cell complex. Both methods lead to the same barcode.

3.3.1 The Elder Rule and the Combinatorial Elder Rule

Let (T, f) be a merge tree. Regarding T as a one-dimensional simplicial complex, we can linearly interpolate the height function from the vertices to the entire tree. The *barcode of the merge tree* (T, f) is the barcode corresponding to the persistence module

$$F : (\mathbb{R}, \leq) \to \mathsf{Vec}_{\mathbb{K}} \qquad \text{where} \qquad F(t) = H_0\Big(f^{-1}\big((-\infty, t]\big)\Big).$$

Although the barcode of F is guaranteed to exist by virtue of Crawley-Boevey's theorem, there is a more direct way of constructing the barcode in the special case of merge trees, called the *Elder rule* [31].

The Elder rule provides a concrete way to compute the barcode of a merge tree via decomposition into branches, i.e., each bar in the barcode corresponds either to a single edge or a list of adjacent edges in the merge tree. According to the Elder rule, each leaf node marks the beginning of a bar in the barcode at the height of the leaf node. If two leaf nodes v_i and v_j such that $f(v_i) > f(v_j)$ share an ancestor at vertex k, the branch that was born “earlier" at v_j survives as it is “elder”, and the branch born v_i dies, creating an interval $[f(v_i), f(v_k))$ in the persistence diagram.

Under this rule, every bar begins at a leaf node and ends at an internal node with the sole exception of the bar that is born at the leaf node with the lowest height, which is paired with infinity. However, in our figures, in keeping with Remark 3.4, the lowest leaf node will be paired with $d_0 = f(r)$, which is the height of the root node when viewed as an embedded finite tree. A simple example is illustrated in Figure 3.2.

The Elder Rule can be defined purely combinatorially for combinatorial merge trees. The output is a set of pairs of labels (the birth and death labels). Recall from Remark 3.10 that a combinatorial merge tree is a combinatorial tree T together with an ordered labelling L_l of the leaves and an ordered labelling L_i of the internal nodes, such that $L_i(v) > L_i(w)$ if v is an ancestor of w. We can define the *combinatorial Elder Rule* in the same way as the Elder Rule is defined in Example 3.3.1 by replacing the function f by the labellings L_l and L_i.

Definition 3.30 (Combinatorial Elder Rule)**.** Let (T, L_l, L_i) be a combinatorial merge tree. As in Example 3.3.1, each leaf node v marks the first coordinate of a pair with the label $L_l(v)$. If two leaf nodes v_i and v_j such that $L_l(v_i) < L_l(v_j)$ share an ancestor v_k, the leaf with the smallest label, v_i gets paired with v_k, creating the pair $(L_l(v_i), L_i(v_k))$. Under this rule, the leaf with the smallest label is paired with the root, which we label by ∞.

The pairing obtained from the combinatorial Elder rule leads to a permutation, where each death label is paired with a birth label. This permutation corresponds to the one of Definition 3.22, see Corollary 3.33.

3.3.2 Topological Morphology Descriptor

The TMD (Topological Morphology Descriptor) [68] carries out the process described above specifically for geometric trees embedded in $\mathbb{R}^3$. It was designed for the study of neuron morphology in [68]. It is a many-to-one function from the set of geometric trees to the set of barcodes,

$$\mathrm{TMD} : \mathcal{T} \to \mathcal{B}$$

that encodes the overall shape of the tree, both the topology of the branching structure of a tree and its embedding in $\mathbb{R}^3$. It is defined recursively as follows.

Let T be a rooted tree with root r and set N of vertices, with subset L of leaves. Let $\delta : N \to \mathbb{R}_{\geq 0}$ be the function that assigns to each vertex its Euclidean distance to the root r. Notice that δ has the required property to be a height function as defined in Definition 3.3.

Intuitively, the output of the TMD algorithm computes the 0-dimensional barcode, or persistence diagram, of the distance function δ. Each bar (b, d) corresponds to a connected component in the sublevel sets $\delta^{-1}\big([0, t)\big)$, that is, a branch of the tree.

Remark 3.31. Note that the birth and death roles are reversed in the TMD algorithm compared to "usual" terminology in persistent homology: the birth corresponds to the bifurcation and the death to the termination of a branch. The endpoints of a bar correspond to the distances to the root from the tip of the branch and from the point where the branch bifurcates from another, longer branch, see Figure 3.9.

Table 6.1 summarizes the terminology used for the TMD algorithm.

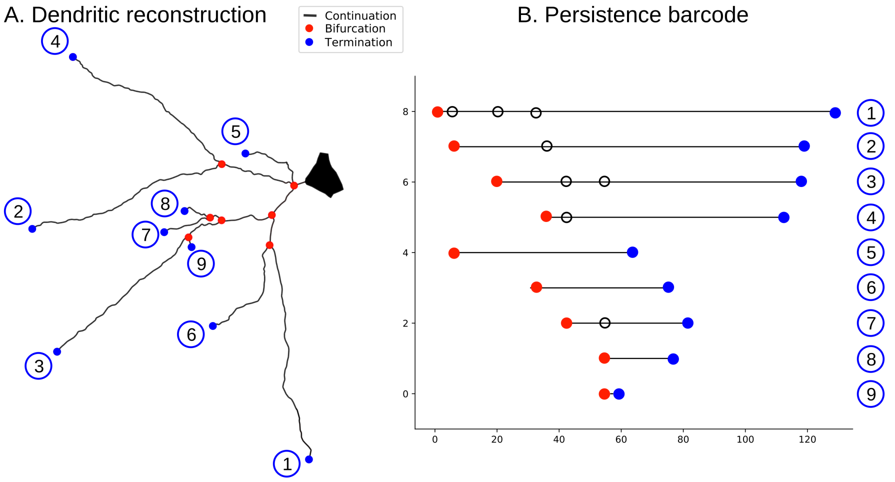


Figure 3.9: (Figure from [68]) The algorithm to encode a neuronal tree structure as a persistence barcode. A. Neuronal tree. B. Persistence barcode generated with TMD. Each branch in the tree (A) corresponds to a bar in the barcode (B); the circled numbers encode the correspondence between branches and bars. Terminations are shown in blue, bifurcations in red, and branches in between in black.

For each $v \in N \setminus L$, let L_v denote the set of leaves of the subtree of T with root at the branch point v. Let $\mu\colon N \to \mathbb{R}$ be the function defined by

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

We order the children of any vertex of T by their μ-value: if $v_1, v_2 \in N$ are siblings, then v_1 is younger than v_2 if $\mu(v_1) < \mu(v_2)$.

The algorithm that extracts the TMD of a geometric tree T proceeds as follows (Figure 3.9). Start by creating a set A of *active vertices*, originally set equal to L, and an empty barcode. For each leaf l, the algorithm proceeds recursively along its unique path to the root r. At each branch point b, one applies the standard Elder Rule [31], removing from A all of the children of b, and adding b to A. One bar is added to the barcode for each child of b except (any one of) the longest. Each child removed from A corresponds to a path from some leaf l to b, which is recorded in the barcode as a bar $\big(\delta(b), \delta(l)\big)$. These operations are applied iteratively to all the vertices until the root r is reached, at which point A contains only r and a leaf l for which μ is maximal among all leaves, which is recorded in the barcode as a bar $\big(0, \delta(l)\big)$.

If T is a digital reconstruction of a neuron, and the function δ is the path distance from the soma, which is the cell body of the neuron and can be seen as the root of the tree, then $\mathrm{TMD}(T)$ is actually a strict barcode. Indeed, the probability for two branch points or leaves to be exactly the same distance from the soma is essentially zero, and $\mathrm{TMD}(T)$ always has a longest bar that contains all the others. This observation justifies our interest in the subset of strict barcodes.

The TMD gives rise to an equivalence relation on $\mathcal{T}$: two geometric trees T and T' are *TMD-equivalent*, denoted $T \underset{\text{tmd}}{\sim} T'$, if $\mathrm{TMD}(T) = \mathrm{TMD}(T')$ as barcodes. We provide in Chapter 6 an in-depth analysis of the TMD-equivalence classes of geometric trees.

3.3.3 Spaces of Trees and Barcodes

In this last section we recall and compare the different notions seen so far. Figure 3.6 and Table 3.1 summarize the important characteristics of combinatorial trees, merge trees, combinatorial phylogenetic trees, and barcodes.

	CT	**MT**	**CMT**	**PT**	**CPT**	**B**
Height function		X		X		
Labels on leaves (births)		X*	X*	X	X	X*
Labels on internal vertices (deaths)		X*	X*			X*
Adjacency	X	X	X	X	X	

Table 3.1: Table summarising the attributes of each object defined in this chapter. CT stands for combinatorial trees (Theorem 3.1), MT for merge trees (Theorem 3.3), CMT for combinatorial merge trees (Theorem 3.9), PT for (metric) phylogenetic trees (Theorem 3.12), CPT for combinatorial phylogenetic trees (Theorem 3.12) and B for barcodes (Theorem 3.17). Labels on leaves and internal vertices of merge trees and combinatorial merge trees are marked by an asterisk to indicate that they are inherited from the height function. Similarly, the "labels" on barcodes (their birth and death values) are inherited from the height function on the tree.

We conclude this section by clarifying the relationship between the two notions of combinatorial equivalence of trees and barcodes that are pertinent to the tree realization problem that we describe in Chapter 5. First, notice that the barcode of a generic merge tree is always strict. The following lemma describes the relationship between the two notions of combinatorial equivalence for trees and barcodes.

Lemma 3.32. *If T and T' are combinatorially equivalent merge trees, then their corresponding barcodes B and B' are combinatorially equivalent as well.*

Proof. Since a tree isomorphism φ as defined in Definition 3.4 preserves both birth and death orders, we need to check only that if the Elder rule pairs the i-th birth node with the j-th death node in T, then the same holds for T'. This is obvious, however, because the unique sequence of edges connecting a pair of nodes in T must be sent to the same sequence of edges connecting these nodes in T', since φ is a graph isomorphism and therefore preserves adjacency relations. □

The corollary below follows directly.

Corollary 3.33. *Let (T, f) be a merge tree, and let (T, L_l, L_i) be its corresponding combinatorial merge tree (Remark 3.10). Let B_T be the barcode of (T, f) obtained by applying the Elder Rule to (T, f), and σ_B its associated permutation. If one applies the combinatorial Elder Rule (Definition 3.30), then one can obtains σ_B from (T, L_l, L_i).*

In other words, the diagram in Figure 3.10 commutes. Figure 3.10 illustrates the relationship between merge trees and their combinatorial equivalence classes and barcodes and their combinatorial equivalence classes, corresponding to permutations.

Considering the Elder rule from a purely combinatorial point of view is very important for the classification of combinatorial merge trees that we describe in Chapter 5.

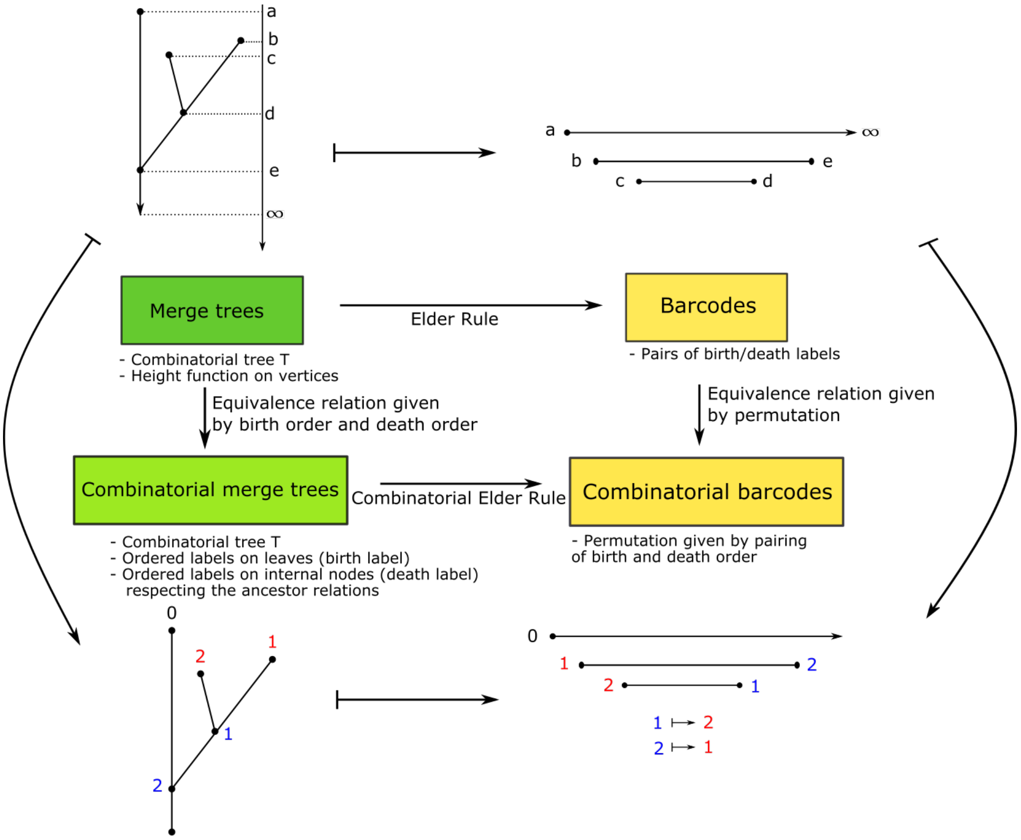

a
b
c
d
e
∞
a
∞
b
e
c
d
Merge trees
Elder Rule
Barcodes
- Combinatorial tree T
- Height function on vertices
- Pairs of birth/death labels
Equivalence relation given by birth order and death order
Equivalence relation given by permutation
Combinatorial merge trees
Combinatorial Elder Rule
Combinatorial barcodes
- Combinatorial tree T
- Ordered labels on leaves (birth label)
- Ordered labels on internal nodes (death label) respecting the ancestor relations
- Permutation given by pairing of birth and death order
0
2
1
1
2
0
1
2
2
1
1 ↦ 2
2 ↦ 1

Figure 3.10: The relationships between merge trees, combinatorial equivalence classes of merge trees, barcodes and permutations. Birth labels are indicated in red, and death labels in blue. The largest bar (corresponding to the essential class) is not taken into account in the combinatorial setting since it is there for every tree/barcode. Therefore we label it by 0. Considering the pairing of the i-th death and the j-th birth given by the combinatorial Elder rule (bottom right) returns the same permutation as the one directly defined from the barcode (top right).

Chapter 4

The Space of Barcodes

In this chapter, we investigate more deeply the space of barcodes and its relation to the symmetric group. Section 4.1 is based on a project that started with L. Kanari and K. Hess in [69] and continued when J. Curry, J. DeSha and B. Mallery joined in [32]. The rest of the chapter extends the equivalence classes of barcodes to new coordinates on the space of barcodes with n bars indexed by the symmetric group. This is joint work with B. Brück.

This chapter focuses on the space of barcodes only, and a barcode need not to come from a tree. Therefore, the essential bar (b_0, d_0) need not exist. We summarize the notation for this chapter in the box below.

Strict barcode: a barcode $B = \{(b_i, d_i)\}_{i \in \{1,\dots,n\}}$ such that $b_i \neq b_i, d_i \neq d_j$ if $i \neq j$. The indexing is arbitrary.
Permutation: Assume that for a strict barcode B one has $b_{i_1} < \ldots < b_{i_n}$ and $d_{j_1} < \ldots < d_{j_n}$. The permutation associated to B is computed via $\tau_b^{-1}\tau_d$, where $\tau_b(k) = i_k$ and $\tau_d(k) = j_k$.
The space $\mathcal{B}_n$: it consists of all barcodes with n bars. Note that in this case, there is no essential bar (b_0, d_0).

4.1 Relations to the Symmetric Group

4.1.1 Combinatorial Barcodes as Permutations

The motivation for this work is to understand the space of barcodes from a combinatorial and geometric point of view. A strict barcode B with n bars can be associated with a permutation $\sigma_B \in \mathrm{Sym}_n$ that tracks the order of the deaths d_i with respect to the order of the births b_i (see Theorem 3.22). This decomposes the set of barcodes with n bars into $n!$ equivalence classes, one for each element of the symmetric group Sym_n. Based on this observation, one can study the combinatorial properties of barcodes by describing these equivalence classes—or equivalently, the elements of Sym_n—and the relations between them.

We can express the relation between barcodes and the symmetric group more concisely as follows. Let $\mathcal{B}_n^{st}$ denote the collection of strict barcodes with n bars. The map that associates to every strict barcode its permutation type defines a bijection between combinatorial equivalence classes of strict barcodes and elements of the symmetric group, i.e.,

$$\mathcal{B}_n^{st}/\sim \qquad \longleftrightarrow \qquad \mathrm{Sym}_n\,.$$

Example 4.1. The relation between the space $\mathcal{B}_4^{st}/\sim$ and the corresponding elements of Sym_4 under the bijection given above is shown in the Cayley graph of Sym_4 in Figure 3.8. Similarly, Figure 4.1 shows all the permutations of Sym_3 and the corresponding persistence diagrams. The Cayley graph of Sym_3 and the space $\mathcal{B}_3^{st}/\sim$ are displayed in Figure 5.5C as well.

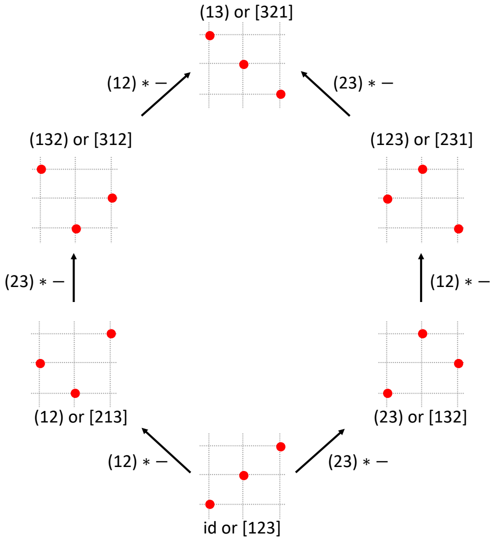


Figure 4.1: Combinatorial equivalence classes of barcodes with three non-essential bars. The associated permutation σ is written next to each diagram in both forms of notation: the image notation is in square brackets, i.e, $[\sigma(1)\sigma(2)\sigma(3)]$, and the cycle notation in parenthesis. The arrows point in the direction of increasing left Bruhat order and exhibit Sym_3 as a poset. Notice that the permutation acts by switching death order of the bars in the barcode

4.1.2 Convexity of Combinatorial Equivalence Classes

In this section we prove that combinatorial equivalence classes are convex in a certain sense: if two strict barcodes B and B' are of the same combinatorial type, then they can be connected by a "line segment" of barcodes[1] all of the same permutation type.

We prove first that the set $\mathcal{B}_n$ admits the algebraic structure necessary to formulate a convexity result.

Lemma 4.2. *1. For all $\lambda \in \mathbb{R}_{>0}$ and $\{(b_i, d_i)\}_{i=1}^n \in \mathcal{B}_n$, the set*

$$\lambda B := \{(\lambda b_i, \lambda d_i)\}_{i=1}^n$$

is also a strict barcode.

2. For all $B = \{(b_i, d_i)\}_{i=1}^n, B' = \{(b_i', d_i')\}_{i=1}^n \in \mathcal{B}_n$, the set

$$B + B' := \{(b_i + b_i', d_i + d_i')\}_{i=1}^n$$

is also a barcode with distinct birth times, which is strict if B and B' have the same permutation type.

Proof. The proof of (1) is trivial, since λ is assumed to be positive, whence multiplication by λ preserves the order of real numbers.

The only subtlety in the proof of (2) concerns distinct death times. If the permutation types of B and B' are different, it could happen that $d_i < d_j$ and $d_i' > d_j'$, but $d_i + d_i' = d_j + d_j'$, so that $B + B'$ would not be strict. If they have the same permutation type, then this cannot happen. □

Lemma 4.3. *For every n and every $\sigma \in \mathrm{Sym}_n$, the set of strict barcodes of permutation type σ is convex, i.e., for B and B' of permutation type σ, the interval*

$$[B, B'] := \{tB + (1-t)B' \mid t \in [0,1]\}$$

is contained in the set of barcodes of permutation type σ.

Proof. Given the previous lemma, it remains only to prove that the permutation type of $tB + (1-t)B'$ is σ, which follows immediately from the observation that

$$d_i < d_j \text{ and } d_i' < d_j' \Longrightarrow d_i + d_i' < d_j + d_j'.$$

□

Remark 4.4. We can also formulate the lemma above as saying that there is a "straight-line path" from B to B',

$$\overline{BB'} : [0,1] \to \mathcal{B}_n : t \mapsto B^t := tB + (1-t)B'.$$

[1] A continuous path of barcodes is sometimes called a *vineyard*. This terminology arises more commonly when barcodes are represented using persistence diagrams, as a path in the space of persistence diagrams is a configuration of paths, some of which enter of exit the diagonal.

It is not hard to show that this function is indeed continuous with respect to both the bottleneck metric and the Wasserstein metric on $\mathcal{B}_n$.

It is interesting also to consider the path $\overline{BB'}$ when the barcodes B and B' are not of the same permutation type. As mentioned in the proof of Lemma 4.2, not every point of $\overline{BB'}$ is necessarily a strict barcode in this case, which allows the path to connect one permutation type to another. One can show that the smallest number of different classes that the path goes through is the length of the shortest path between the two corresponding permutations of B and B' on the Cayley graph defined using the generating set of elementary (neighboring) transpositions $\tau_i = (i, i+1)$. This value is related to the Bruhat order. A more complete explanation involves describing the space of barcodes in terms of a family of convex sets that fiber over the dual of the permutohedron, which is the purpose of the next section.

Example 4.5. Figure 4.2 shows an example of the path described in the proof above, using the representation of barcodes as persistence diagrams. The path consists of the straight lines between the matched points of the diagrams. Note that the dotted lines indicating the births and deaths never cross for the same birth and death order, respectively, because the barcodes stay in the same permutation class at each step of the path. It *is* possible for b_1 to be greater than b_2', for example, but the relative order of births and deaths does not change.

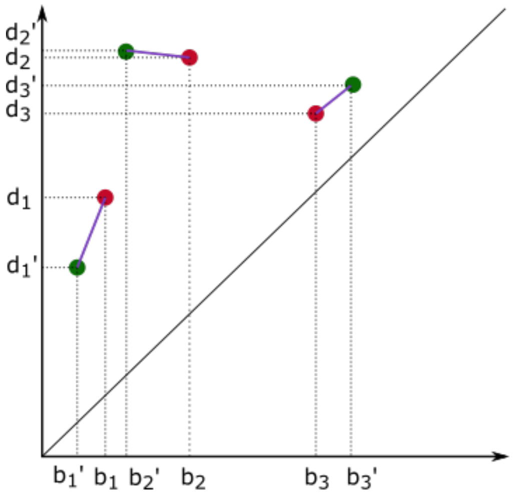


Figure 4.2: Continuous path between two barcodes in the same combinatorial class. We show the persistence diagrams of each barcode. The first one B is indicated by red dots and the second one by green dots, and the path B^t is in purple.

Lemma 4.3 allows us to fix a standardized representative of each combinatorial barcode type, making the connection to the symmetric group explicit.

Definition 4.6. A barode B is in *standard form* if there is a permutation σ of the set $\{1, \ldots, n\}$ so that

$$B = \{(i, \sigma(i) + n)\}_{i \in \{1,\ldots,n\}}.$$

It is clear that B is strict and has permutation type σ. We sometimes write $B(\sigma)$ for the standard barcode associated to σ.

Lemma 4.3 implies that any strict barcode B of permutation type σ can be connected via a straight-line path to the barcode $B(\sigma)$.

4.2 Stratifying the space of barcodes using Coxeter complexes

Considering the Cayley graph of the symmetric group with respect to the generating set given by adjacent transpositions $(i, i+1)$ yields a combinatorial representation of the elements of Sym_n. It tells us how a pair of permutations can be transformed into one another using transpositions one step at a time. However, it yields no information about "higher order relations" that exist among larger sets of permutations and it does not offer a way to continuously change permutations or the associated equivalence classes of barcodes.

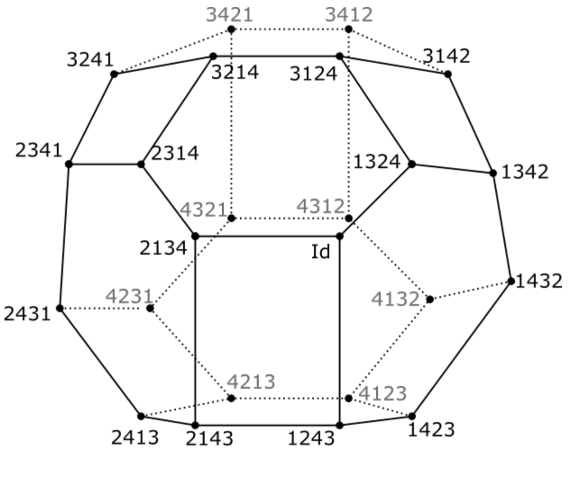


Figure 4.3: The permutohedron [100] of order 4 is a polyhedral decomposition of the sphere where each vertex corresponds to an element of the symmetric group Sym_4. Its 1-skeleton is the Cayley graph of Sym_4 (see Figure 3.8).

A way to resolve this is to add higher dimensional cells to the Cayley graph and to consider it more geometrically as a cell complex instead of as a (combinatorial) graph. A first approach would be to use that the Cayley graph of Sym_n is the 1-skeleton of the permutohedron [100] of order n, see Figure 4.3. This observation embeds the Cayley graph into a polyhedral decomposition of the $(n-2)$-sphere. As this is a more geometric object, it allows to continuously "walk" from one permutation to another. The problem is that only the vertices (and not the higher dimensional cells) of the permutohedron have an interpretation in terms of elements of the symmetric group. This makes it unclear how such a walk would continuously change one permutation into an other. Furthermore, this representation lacks a notion of "size" for barcodes. For instance, the two barcodes depicted in Figure 4.4 lie in the same equivalence class and hence have the same associated permutation.

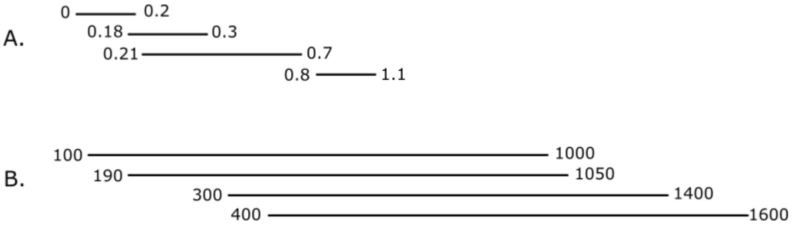


Figure 4.4: Two barcodes with the same associated permutation (the identity [1234]) but with large differences in their birth and death values.

The alternative that we suggest to overcome these problems is to work with Coxeter complexes (see Section 2.3.2) instead of permutohedra. The Coxeter complex associated to Sym_n is the dual of the permutohedron of order n (Figure 4.5). It forms a simplicial decomposition of the $(n-2)$-sphere and is well-studied in the context of reflection groups and Tits buildings. For us, it has the advantage that its top-dimensional simplices correspond in a natural way to permutations and only passing through a face of lower dimension changes such a permutation. This allows for a better description of continuous changes between different permutations. It also has the advantage that it comes with an embedding in $\mathbb{R}^n$, where the additional two real parameters that are needed to describe positions relative to this $(n-2)$-dimensional space have a natural interpretation in terms of the "size" of barcodes.

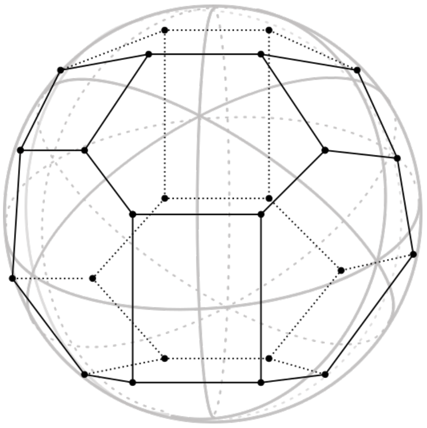


Figure 4.5: The permutohedron of order 4 (black) is the dual of the Coxeter complex $\Sigma(\mathrm{Sym}_4)$ (grey).

In this section, we use Coxeter complexes to develop a description of the set $\mathcal{B}_n$ of barcodes with n bars with coordinates that have natural interpretation when doing statistics with barcodes. These coordinates define a stratification of $\mathcal{B}_n$ where the highest dimensional strata are indexed by the symmetric group. The main results of this section can be summarized as follows:

Theorem 4.7. *Let $\mathcal{B}_n$ denote the set of barcodes with n bars.*

1. *$\mathcal{B}_n$ can in a natural way be seen as a subset of a quotient $\mathrm{Sym}_n \backslash \mathbb{R}^{2n}$.*
2. *$\mathcal{B}_n$ is stratified over the poset of marked double cosets of parabolic subgroups of Sym_n.*
3. *Using this description, one obtains a decomposition of $\mathcal{B}_n$ into different regions. Each region is characterized as the set of all barcodes having the same average birth and death, the same standard deviation of births and deaths and the same permutation type $\sigma_B \in \mathrm{Sym}_n$.*
4. *This description gives rise to metrics on $\mathcal{B}_n$ that coincide with modified versions of the bottleneck and Wasserstein metrics.*

For more detailed and formal statements of these results, see Theorem 4.14, Theorem 4.21, Theorem 4.22 and Theorem 4.24.

To obtain this description of $\mathcal{B}_n$, we proceed as follows. A barcode is an (unordered) multiset of n pairs of real numbers (birth and death times). It can hence be seen as a point in $\mathrm{Sym}_n \backslash \mathbb{R}^n \times \mathbb{R}^n$, where the action of Sym_n permutes the coordinate pairs. As for every barcode the birth is smaller then the death, $\mathcal{B}_n$ is only a proper subset of this quotient of $\mathbb{R}^{2n}$. Let $\Sigma(\mathrm{Sym}_n)$ denote the Coxeter complex for Sym_n. It is homeomorphic to an $(n-2)$-sphere, so we can decompose $\mathbb{R}^n$ as

$$\mathbb{R}^n \cong \mathrm{cone}(\Sigma(\mathrm{Sym}_n)) \times \mathbb{R},$$

where $\mathrm{cone}(\Sigma(\mathrm{Sym}_n)) = \Sigma(\mathrm{Sym}_n) \times [0,\infty)/(x,0) \sim (y,0) \cong \mathbb{R}^{n-1}$. This decomposition yields coordinates $x_\theta, \bar{x}, \|v_x\|$, where x_θ specifices a point on the Coxeter complex, $\|v_x\|$ is the "cone parameter" and $\bar{x}$ parametrizes the remaining $\mathbb{R}$ (for details, see Theorem 4.9, where the naming becomes clear as well). In summary, this allows one to describe $\mathcal{B}_n$ as a subset of

$$\mathcal{B}_n \subset \mathrm{Sym}_n \backslash [\mathrm{cone}(\Sigma(\mathrm{Sym}_n)) \times \mathbb{R} \times \mathrm{cone}(\Sigma(\mathrm{Sym}_n)) \times \mathbb{R}].$$

We call the coordinates that we obtain from this description *Coxeter coordinates.* It turns out that for each barcode, these coordinates are $b_\theta, \bar{b}$, $\|v_b\|$ and $d_\theta, \bar{d}$, $\|v_d\|$, where $\bar{b}$ and $\bar{d}$ are the averages of the births and deaths, $\|v_b\|$ and $\|v_d\|$ are their standard deviations, and the coordinates b_θ and d_θ describe the permutation equivalence class of the barcode (Theorem 3.22). The stratification one obtains is induced by the simplicial structure of $\Sigma(\mathrm{Sym}_n)$.

The advantages of these new coordinates are two-fold. Firstly, one obtains coordinates that uniquely specify barcodes and are yet compatible with the combinatorial structure of $\mathcal{B}_n$ given by permutation equivalence classes. Secondly, one resolves the earlier-mentioned problem that permutation equivalence classes themselves carry no notion of "size". The decomposition of $\mathcal{B}_n$ into regions subdivides these equivalence classes by also taking into account the averages and standard deviations of births and deaths. This makes these regions a finer invariant than the permutation type. Therefore, they offer a new way to study statistics of barcodes by taking into account both the average and standard deviation of births and deaths, which are commonly used summaries in TDA, and permutation statistics tools, such as the number of descents and inversion numbers, or the tree realization number that we proved useful for the study of inverse problem for barcodes and trees in Chapters 5 and 6.

This section is based on joint work with B. Brück.

4.2.1 Coxeter complex coordinates on $\mathbb{R}^n$

In this section, we describe $\mathbb{R}^n$ as the product of a cone over the Coxeter complex $\Sigma(\mathrm{Sym}_n)$ with a 1-dimensional space orthogonal to it. This description is based on a standard way for realising Sym_n as a reflection group [1, Example 1.11]. In terms of Coxeter groups, this is often called the "dual representation", see e.g. [1, Section 2.5.2]. Theorem 4.11 below goes through the following steps in detail for the case $n = 3$.

In what follows, we will consider $\mathbb{R}^n$ with the l^2-norm $\|\cdot\|$ that is induced by the standard scalar product $\langle\cdot,\cdot\rangle$. We let $e_1, \ldots, e_n$ denote the standard basis.

The symmetric group Sym_n acts on $\mathbb{R}^n$ by permuting this standard basis. This action can be expressed in coordinates as

$$\gamma \cdot (x_1, \ldots, x_n) = (x_{\gamma^{-1}(1)}, \ldots, x_{\gamma^{-1}(n)}). \tag{4.1}$$

It is norm-preserving and fixes the 1-dimensional subspace $L = \langle e \rangle$ spanned by $e := e_1 + \cdots + e_n = (1, \ldots, 1)$. Hence, there is an induced action on the orthogonal complement $V = e^{\perp}$, which can be described as

$$V = \{(x_1, \ldots, x_n) \in \mathbb{R}^n \,|\, \Sigma_{i=1}^n x_i = 0\}.$$

Note that L is the subspace consisting of all $(x_1, \ldots, x_n) \in \mathbb{R}^n$ where $x_i = x_j$ for all i, j. So in particular, every $(x_1, \ldots, x_n) \in \mathbb{R}^n \setminus L$ has at least two coordinates that are different from one another.

The subspace V has a natural structure of a cone over the Coxeter complex $\Sigma(\mathrm{Sym}_n)$ associated to Sym_n. The transposition $(i, j) \in \mathrm{Sym}_n$ acts on V by orthogonal reflection along the hyperplane

$$\{(x_1, \ldots, x_n) \in \mathbb{R}^n \,|\, x_i = x_j\},$$

permuting the i-th and j-th coordinates. Let $\mathcal{H}$ be the collection of all these hyperplanes, and let S_r denote the $(n-2)$-sphere of radius $r > 0$ centered at the origin in V (with respect to the norm induced by the restriction of the standard scalar product on $\mathbb{R}^n$), i.e., $S_r = \{v \in V \mid \|v\| = r\}$.

Lemma 4.8 ([1, Examples 1.10, 1.4.7 & 1.81])**.** *The hyperplanes $\mathcal{H}$ induce a triangulation of S_r. The resulting simplicial complex Σ is isomorphic to the Coxeter complex $\Sigma(\mathrm{Sym}_n)$ as Sym_n-spaces.*

The set of points $x \in \mathbb{R}^n$ such that all coordinates are different is the *configuration space*

$$\mathrm{Conf}_n(\mathbb{R}) = \{(x_1, \ldots, x_n) \in \mathbb{R}^n \mid i \neq j \implies x_i \neq x_j\}.$$

The previous lemma describes how a permutation in Sym_n can be associated to each point $x \in \mathrm{Conf}_n(\mathbb{R})$. To understand why this is true, observe that if C is a connected component of $S_r \setminus \bigcup \mathcal{H}$, then for all $(x_1, \ldots, x_n) \in C$:

- if $i \neq j$, then $x_i \neq x_j$, i.e., $(x_1, \ldots, x_n) \in \mathrm{Conf}_n(\mathbb{R})$;
- if $(y_1, \ldots, y_n) \in C$, then $y_i < y_j$ if and only if $x_i < x_j$.

In particular, there is a unique $\tau \in \mathrm{Sym}_n$ such that

$$(x_1, \ldots, x_n) \in C \iff x_{\tau(1)} < x_{\tau(2)} < \cdots < x_{\tau(n)}. \tag{4.2}$$

In other words, the order of the elements $x_1, \ldots, x_n$ is given by $\tau((1, \ldots, n))$. See Figure 2.3 above for the case $n = 4$. The connected components of $S_r \setminus \bigcup \mathcal{H}$ are exactly the (interiors of) the maximal simplices of Σ. Sending each such component C to the facet of $\Sigma(\mathrm{Sym}_n)$ that corresponds to the permutation τ defined by Equation 4.2 gives the desired isomorphism $\Sigma \cong \Sigma(\mathrm{Sym}_n)$.

Using spherical coordinates, we can express every point $v \in V$ in terms of a radial component $r > 0$ and an angular component, which is equivalent to specifying a point $v_\theta \in S_r$ (i.e., a point in the geometric realization of $\Sigma(\mathrm{Sym}_n)$). The upshot of this is that we obtain a new set of coordinates for points in $\mathbb{R}^n \setminus L$.

Proposition 4.9. *Let $n \geq 2$. There exist two projection maps*

$$p : \mathbb{R}^n \longrightarrow \mathbb{R} \times \mathbb{R}_{\geq 0} : x \mapsto, (\bar{x}, \|v_x\|),$$

where $\bar{x} = \frac{1}{n}\sum_{i=1}^n x_i$ and $\|v_x\| = \left(\sum_{i=1}^n |x_i - \bar{x}|^2\right)^{1/2}$, and

$$q : \mathbb{R}^n \setminus L \longrightarrow \Sigma(\mathrm{Sym}_n)$$

that define a bijection

$$(p\big|_{\mathbb{R}^n \setminus L}, q) : \mathbb{R}^n \setminus L \longrightarrow \mathbb{R} \times \mathbb{R}_{>0} \times \Sigma(\mathrm{Sym}_n).$$

Let Sym_n act on $\mathbb{R}^n$ by permuting the coordinates (Equation 4.1) and on the product $\mathbb{R} \times \mathbb{R}_{>0} \times \Sigma(\mathrm{Sym}_n)$ by extending the action on $\Sigma(\mathrm{Sym}_n)$ trivially on the first two factors. Then the map $(p\big|_{\mathbb{R}^n \setminus L}, q)$ is Sym_n-equivariant.

Proof. For every $x \in \mathbb{R}^n$, the orthogonal decomposition $\mathbb{R}^n = \langle e \rangle \oplus V$ gives a unique way to write $x = \bar{x} \cdot e + v_x$ with $\bar{x} \in \mathbb{R}$ and $v_x \in V$, where

$$\bar{x} = \frac{\langle e, x \rangle}{\langle e, e \rangle} = \sum_{i=1}^n x_i/n = \frac{1}{n}\sum_{i=1}^n x_i.$$

We can describe the projection $v_x = x - \bar{x} \cdot e \in V$ in spherical coordinates. Its norm (the radius of the sphere) is

$$\|v_x\| = \|x - \bar{x} \cdot e\| = \left(\sum_{i=1}^n |x_i - \bar{x}|^2\right)^{1/2},$$

so v_x is determined by this value together with a point x_θ on the $(n-2)$-sphere $S_{\|v_x\|}$, or equivalently on the geometric realization of $\Sigma(\mathrm{Sym}_n)$. Notice that $x \in L$ if and only if $v_x = 0$, as the line L intersects V at its origin, in which case the choice of x_θ is not unique.

We define the map $p : \mathbb{R}^n \longrightarrow \mathbb{R} \times \mathbb{R}_{\geq 0} : x \mapsto (\bar{x}, \|v_x\|)$ and the map $q : \mathbb{R}^n \setminus L \longrightarrow S^{n-2} : x \mapsto x_\theta$. The point x_θ is well-defined since $x \notin L$ and therefore there exist i, j such that $x_i \neq x_j$. It is easy to see that $(p|_{\mathbb{R}^n \setminus L}, q)$ is a bijection, i.e., that given $c_1 \in \mathbb{R}$, $c_2 \in \mathbb{R}_{>0}$ and $c_3 \in \Sigma(\mathrm{Sym}_n)$, there is a unique $x \in \mathbb{R}^n \setminus L$ such that $c_1 = \bar{x}$, $c_2 = \|v_x\|$ and $c_3 = x_\theta$.

The fact that $(p|_{\mathbb{R}^n \setminus L}, q)$ is Sym_n-equivariant follows from Theorem 4.8 and because permuting the coordinates of $x \in \mathbb{R}^n$ changes neither the average $\frac{1}{n}\sum_i x_i$ nor the standard deviation $\left(\sum_i |x_i - \bar{x}|^2\right)^{1/2}$. □

To summarize, every point $x = (x_1, \ldots, x_n) \in \mathbb{R}^n \setminus L$ determines the following three data:

1. its projection to L, given by $\bar{x} = \frac{1}{n}\sum_{i=1}^n x_i \in \mathbb{R}$;
2. the norm of its projection to V, given by $\|v_x\| = \left(\sum_{i=1}^n |x_i - \bar{x}|^2\right)^{1/2} \in \mathbb{R}_{>0}$;
3. a point x_θ in the geometric realization of the Coxeter complex $\Sigma(\mathrm{Sym}_n)$ associated to Sym_n.

Furthermore, x is uniquely determined by these three coordinates.

Remark 4.10. Since $\mathbb{R}^n = \mathbb{R}^n \setminus L \sqcup L$ and $\mathbb{R}_{>0} \times \Sigma(\mathrm{Sym}_n) \cong \mathrm{cone}(\Sigma(\mathrm{Sym}_n)) \setminus \{*\}$, the map above gives rise to a decomposition $\mathbb{R}^n \cong \mathrm{cone}(\Sigma(\mathrm{Sym}_n)) \times \mathbb{R}$. Indeed, the line $L \subset \mathbb{R}^n$ corresponds to points $x \in \mathbb{R}^n$ with $v_x = 0$, which could be seen as “spheres of radius 0” in the projection q.

Example 4.11. We go through the previous construction in detail for the case of $\mathbb{R}^3$ equipped with the natural action of the symmetric group Sym_3, illustrating the example in Figure 4.6.

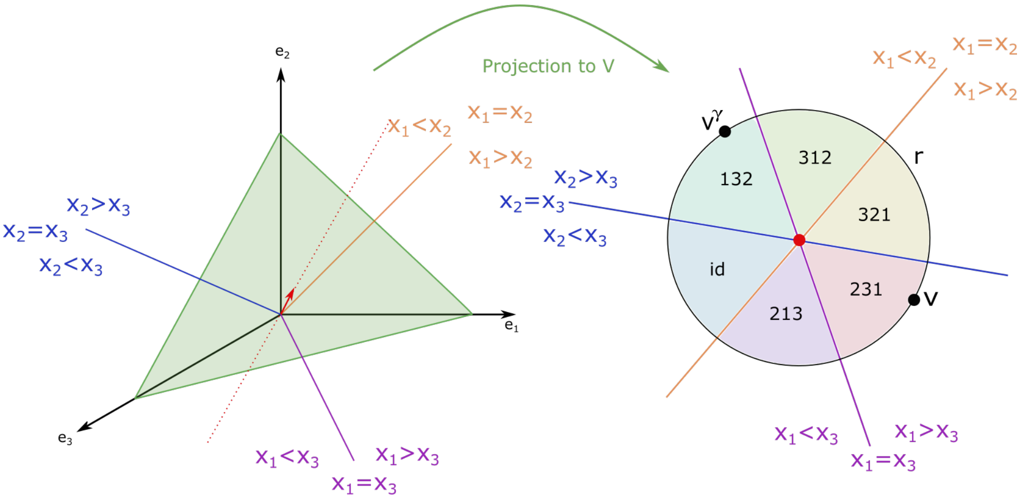


Figure 4.6: Example of the decomposition of $\mathbb{R}^3$ in Coxeter coordinates.

Consider $\mathbb{R}^3 = \langle e_1, e_2, e_3 \rangle$. The symmetric group Sym_3 acts on $\mathbb{R}^3$ by permuting the coordinates of each vector (x_1, x_2, x_3):

$$\gamma \cdot (x_1, x_2, x_3) = (x_{\gamma^{-1}(1)}, x_{\gamma^{-1}(2)}, x_{\gamma^{-1}(3)}).$$

Each $\gamma \in \mathrm{Sym}_3$ can be written as a product of transpositions (i, j), and its action on $\mathbb{R}^3$ is given by the performing the corresponding sequence of reflections along the hyperplanes $x_i = x_j$. The three (2-dimensional) planes corresponding to the equations $x_1 = x_2$, $x_2 = x_3$ and $x_1 = x_3$ are indicated as lines on the left hand side of Figure 4.6 to make the picture clearer. The subspace L that is invariant under this action is spanned by the vector $(1, 1, 1) = e$, shown in red in Figure 4.6.

We can define new coordinates on $\mathbb{R}^3$, lying in $\langle e \rangle = L$ and $e^\perp = V$, a 2-dimensional subspace whose affine shift is depicted in green in Figure 4.6, reflecting the decomposition of $\mathbb{R}^3$ into a product of $\langle e \rangle$ and V. A point $x \in \mathbb{R}^3$ can now be written as $\bar{x} \cdot e + v_x$, where $\bar{x} \in \mathbb{R}$ and $v_x \in V$.
We show on the right hand side of Figure 4.6 how V, represented as $\mathbb{R}^2$, has the structure of a cone over a Coxeter complex. The figure shows the projections of the planes $x_1 = x_2$, $x_2 = x_3$ and $x_1 = x_3$ and the intersection of V with the subspace $\langle e \rangle$ (red dot). To obtain the cone structure on V, we give it spherical coordinates (i.e., polar coordinates in this case). The first coordinate is the radius r, which determines a 1-sphere centred at the origin (the black circle). On the circle, a point v_x is determined by an angle x_θ. Intersecting the circle with the hyperplanes, we decompose it into $|\mathrm{Sym}_3| = 6$ (coloured) strata indexed by the symmetric group and forget about the angle x_θ. For instance, if $v = (v_1, v_2, v_3)$ with $v_2 < v_3 < v_1$, the point v lies in the stratum indexed by [231]; this is the unique region that lies on those sides of the hyperplanes that satisfy $x_1 > x_2$, $x_2 < x_3$ and $x_1 > x_3$.

Let $\gamma = (12)$. It acts on v via $\gamma \cdot v = (v_{\gamma^{-1}(1)}, v_{\gamma^{-1}(2)}, v_{\gamma^{-1}(3)}) = (v_2, v_1, v_3)$. We denote its image by $v^\gamma := \gamma \cdot v$. The order of the coordinates of v^γ satisfies $v_1^\gamma \leq v_3^\gamma \leq v_2^\gamma$, so v^γ lies in the stratum indexed by the permutation [132]. The image v^γ of v through the action of γ corresponds to the reflection through the hyperplane $x_1 = x_2$.

Remark 4.12. There are two special cases in Theorem 4.9, when $x_i = x_j$ for all i, j, i.e., $(x_1, \dots, x_n) \in L$ and when $x_i \neq x_j$ for all $i \neq j$, i.e., $(x_1, \dots, x_n) \in \mathrm{Conf}_n(\mathbb{R})$. For the former, we have $p(x) = (\bar{x}, \|v_x\|) = (x_i, 0)$ and x_θ is not defined. For the latter, $q(x) = x_\theta$ lies in the interior of a top-dimensional simplex of $\Sigma(\mathrm{Sym}_n)$. Hence, it determines a *unique* element $\tau_x \in \mathrm{Sym}_n$. In fact, these are just the two extremes of a family of situations that can occur.

If $x \in \mathrm{Conf}_n(\mathbb{R})$, then x_θ lies on an intersection of hyperplanes in $\mathcal{H}$ and hence on a lower-dimensional face of $\Sigma(\mathrm{Sym}_n)$. There exists a permutation $\tau \in \mathrm{Sym}_n$ such that

$$x_{\tau(1)} \leq x_{\tau(2)} \leq \cdots \leq x_{\tau(n)},$$

but τ is not unique. It is defined only up to multiplication by elements of the subgroup

$$P = \left\{\gamma \in \mathrm{Sym}_n \,\middle|\, x_{\tau(i)} = x_{\tau\gamma(i)} \text{ for all } i\right\}.$$

Note that P is generated by adjacent transpositions $(i, i+1)$, i.e., it is of the form $\langle T\rangle$, where $T \subset S$ is a subset of the set S of simple reflections of Sym_n. Hence, it is a parabolic subgroup of Sym_n. The number of adjacent transpositions generating P depends on how many coordinates of $(x_1, \dots, x_n)$ agree, or, equivalently, the number of hyperplanes in $\mathcal{H}$ it lies on. Intuitively speaking, one could phrase this as "the more of the x_i's take the same value, the less 'permutation information' is left". The coset

$$\tau P = \{\rho \in \mathrm{Sym}_n \mid x_{\rho(1)} \leq \dots \leq x_{\rho(n)}\},$$

corresponds to the lowest dimensional face of $\Sigma(\mathrm{Sym}_n)$ that x lies on. It depends only on the order of the values of the x_i, not on the choice of τ. If $x \in L$, then $\tau P = \mathrm{Sym}_n$. This could be interpreted as the degenerate case where x_θ lies on the unique (-1)-dimensional face of $\Sigma(\mathrm{Sym}_n)$ (see Theorem 2.23).

4.2.2 Coxeter coordinates for the space of barcodes

Describing $\mathcal{B}_n$ as a quotient

In this section, we describe $\mathcal{B}_n$ as a subset of a quotient of $\mathbb{R}^{2n}$. This will be used in the next section to equip this space with Coxeter complex coordinates.

Let $X := \mathrm{Sym}_n \backslash \mathbb{R}^n \times \mathbb{R}^n$, where Sym_n acts diagonally by permuting the coordinates, i.e., for $\gamma \in \mathrm{Sym}_n$, we set

$$\gamma \cdot (x_1, \dots, x_n, y_1, \dots, y_n) = (x_{\gamma^{-1}(1)}, \dots, x_{\gamma^{-1}(n)}, y_{\gamma^{-1}(1)}, \dots, y_{\gamma^{-1}(n)}).$$

The elements of X are equivalence classes of tuples $(x_1, \dots, x_n, y_1, \dots, y_n) \in \mathbb{R}^n \times \mathbb{R}^n$, which are denoted by $[x_1, \dots, x_n, y_1, \dots, y_n]$.

Remark 4.13. We write $X := \mathrm{Sym}_n \backslash \mathbb{R}^n \times \mathbb{R}^n$ to emphasize that Sym_n acts from the left on this space. The reason we stress this is that later on, we will combine the statements here with descriptions of the Coxeter complex. There, the simplices are given by cosets τP, and the symmetric group acts on them by *left* multiplication.

There is a map ϕ from the space of barcodes with n bars to X given by

$$\begin{aligned} \phi : \mathcal{B}_n &\to X = \mathrm{Sym}_n \backslash \mathbb{R}^n \times \mathbb{R}^n \\ \{(b_i, d_i)\}_{i \in \{1, \dots, n\}} &\mapsto [b_1, \dots, b_n, d_1, \dots, d_n]. \end{aligned}$$

The image of ϕ is independent of the choice of indices for the bars of the barcode because the action of Sym_n is factored out. The map ϕ is clearly injective, but it is not surjective as the birth time of a homology class is always smaller than its death time, i.e., $b - i < d - i$ for all i. The image of ϕ is the subspace Y of X given by

$$Y \coloneqq \mathrm{Sym}_n \setminus \{(x_1, \dots, x_n, y_1, \dots, y_n) \in \mathbb{R}^n \times \mathbb{R}^n \,|\, x_i < y_i \text{ for all } i\}\,.$$

For later reference, we note this observation in the following.

Proposition 4.14. *The map ϕ defines a bijection $\mathcal{B}_n \to Y \subset \mathrm{Sym}_n \backslash \mathbb{R}^n \times \mathbb{R}^n$.*

In Section 4.2.4, we equip $\mathcal{B}_n$ with metrics inspired by the bottleneck and Wasserstein distances. The map ϕ is an isometry with respect to these metrics.

Coxeter complexes for birth and death

We now introduce the Coxeter complex coordinates for $\mathcal{B}_n$. These coordinates are obtained by applying the map $(p\big|_{\mathbb{R}^n \setminus L}, q)$ of Theorem 4.9 to the two copies of $\mathbb{R}^n$ in Y.

Theorem 4.15. *Every barcode $\{(b_i, d_i)\}_{i \in \{1,\dots,n\}} \in \mathcal{B}_n$ such that at least two of the b_i and two of the d_i are different from each other determines the following five data:*

1. *its* average birth time $\bar{b} = \sum_{i=1}^n b_i/n \in \mathbb{R}$*;*
2. *its* average death time $\bar{d} = \sum_{i=1}^n d_i/n \in \mathbb{R}$*;*
3. *its* birth standard deviation $\|v_b\| = \left(\sum_{i=1}^n |b_i - \bar{b}|^2\right)^{1/2} \in \mathbb{R}_{>0}$*;*
4. *its* death standard deviation $\|v_d\| = \left(\sum_{i=1}^n |d_i - \bar{d}|^2\right)^{1/2} \in \mathbb{R}_{>0}$*;*
5. *an orbit* $\mathrm{Sym}_n \cdot (b_\theta, d_\theta) \in \mathrm{Sym}_n \backslash \Sigma(\mathrm{Sym}_n) \times \Sigma(\mathrm{Sym}_n)$.

Furthermore, these five data uniquely determine B.

Proof. Let $B = \{(b_i, d_i)\}_{i \in \{1,\dots,n\}}$ be such that at least two b_i and two d_i are different. By assumption, both $(b_1, \dots, b_n)$ and $(d_1, \dots, d_n)$ are points in $\mathbb{R}^n \setminus L$. The image of B under ϕ (Theorem 4.14) is

$$\phi(B) = [b_1, ..., b_n, d_1, ..., d_n] \in \mathrm{Sym}_n \backslash (\mathbb{R}^n \setminus L \times \mathbb{R}^n \setminus L).$$

Since the map $(p\big|_{\mathbb{R}^n \setminus L}, q)$ is Sym_n-equivariant (Theorem 4.9), it induces a bijection

$$\mathrm{Sym}_n \backslash \big(\mathbb{R}^n \setminus L \times \mathbb{R}^n \setminus L\big) \cong \mathrm{Sym}_n \backslash \big(\mathbb{R} \times \mathbb{R}_{>0} \times \Sigma(\mathrm{Sym}_n)) \times (\mathbb{R} \times \mathbb{R}_{>0} \times \Sigma(\mathrm{Sym}_n))\big).$$

The image of $[b_1, ..., b_n, d_1, ..., d_n]$ under this bijection is the Sym_n-orbit of

$$(p\big|_{\mathbb{R}^n \setminus L}, q)^2(b_1, ..., b_n, d_1, ..., d_n) = (\bar{b}, \|v_b\|, b_\theta, \bar{d}, \|v_d\|, d_\theta).$$

The claim now follows since the action of Sym_n on $(\bar{b}, \|v_b\|, b_\theta, \bar{d}, \|v_d\|, d_\theta)$ is trivial on $\bar{b}$, $\|v_b\|$, $\bar{d}$, $\|v_d\|$ and is given by the action of Sym_n on the Coxeter complex $\Sigma(\mathrm{Sym}_n)$ for b_θ, d_θ. □

4.2.3 A stratification of $\mathcal{B}_n$

In this section, we describe the stratification that we obtain from the description of $\mathcal{B}_n$ in terms of Coxeter complexes.

We start by extending Theorem 3.22, the permutation assigned to a strict barcode, to the general case of $\mathcal{B}_n$. For non-strict barcodes, we cannot uniquely assign a permutation. However, there is a nice description of the set of all possible such permutations in terms of double cosets of parabolic subgroups:

Definition 4.16. For a barcode $B = \{(b_i, d_i)\}_{i\in\{1,\dots,n\}} \in \mathcal{B}_n$, let τ_b and τ_d be elements of Sym_n such that $b_{\tau_b(1)} \leq \ldots \leq b_{\tau_b(n)}$ and $d_{\tau_d(1)} \leq \ldots \leq d_{\tau_d(n)}$. Let

$$P_b^B = \left\{\gamma \in \mathrm{Sym}_n \,\middle|\, b_{\tau_b(i)} = b_{\tau_b\gamma(i)}\, \forall\, i\right\},\ P_d^B = \left\{\gamma \in \mathrm{Sym}_n \,\middle|\, d_{\tau_d(i)} = d_{\tau_d\gamma(i)}\, \forall\, i\right\}.$$

The *double coset D_B associated to B* is defined as $D_B := P_b^B \tau_b^{-1}\tau_d P_d^B$.

Remark 4.17. Note that while τ_b and τ_d depend on the ordering of the barcode, P_b^B and P_d^B do not. The groups P_b^B and P_d^B are parabolic subgroups of Sym_n, as was observed in Theorem 4.12. The cosets

$$\tau_b P_b^B = \{\rho \in \mathrm{Sym}_n \mid b_{\rho(1)} \leq \ldots \leq b_{\rho(n)}\}$$

and

$$\tau_d P_d^B = \{\rho \in \mathrm{Sym}_n \mid d_{\rho(1)} \leq \ldots \leq d_{\rho(n)}\},$$

which are the sets of permutations that preserve the order of the b_i and d_i respectively, do not depend on the indexing of B either. Hence, the double coset $D_B = (\tau_b P_b^B)^{-1} \cdot \tau_d P_d^B$ is indeed an invariant of the barcode B. Furthermore, if B is a strict barcode, then $P_b^B = \{\mathrm{id}\} = P_d^B$, so $D_B = \left\{\tau_b^{-1}\tau_d\right\} = \{\sigma_B\}$ and we recover Theorem 3.22.

Example 4.18. Let

$$B = \{(b_1, d_1) = (1, 10), (b_2, d_2) = (2, 5), (b_3, d_3) = (4, 5), (b_4, d_4) = (4, 7)\} \in \mathcal{B}_4.$$

One has $b_1 < b_2 < b_3 = b_4$ and $d_2 = d_3 < d_4 < d_1$. Let $\tau_b = [1234]$ and $\tau_d = [2341]$. They satisfy $b_{\tau_b(1)} \leq \ldots \leq b_{\tau_b(4)}$ and $d_{\tau_d(1)} \leq \ldots \leq d_{\tau_d(4)}$ respectively, but so do $\tau_b' = [1243]$ and $\tau_d' = [3241]$. In this case, one has $P_b^B = \{\mathrm{id}, (34)\}$, $P_d^B = \{\mathrm{id}, (12)\}$ and $\tau_b P_b^B = \{[1234], [1243]\}$, $\tau_d P_d^B = \{[2341], [3241]\}$. The double coset

$$\begin{aligned} D_B &= \{\gamma_b \tau_b^{-1} \tau_d \gamma_d \mid \gamma_b \in P_b^B, \gamma_d \in P_d^B\} \\ &= \{\tau_b^{-1}\tau_d, \tau_b'^{-1}\tau_d, \tau_b^{-1}\tau_d', \tau_b'^{-1}\tau_d'\} \\ &= \{[2341], [2431], [3241], [4231]\} \end{aligned}$$

is the set of all the permutations σ that satisfy that the j-th death (in increasing order) is paired with the $\sigma(j)$-th birth.

Recall that the Coxeter complex $\Sigma(\mathrm{Sym}_n)$ is a simplicial complex with simplices given by cosets of parabolic subgroups. This simplicial decomposition gives $\Sigma(\mathrm{Sym}_n)$ the structure of a stratified space over the poset of cosets of parabolic subgroups equipped with reverse inclusion. Taking the cone and products of these simplices yields a decomposition of

$$\mathbb{R}^{2n} \cong \mathrm{cone}(\Sigma(\mathrm{Sym}_n)) \times \mathbb{R} \times \mathrm{cone}(\Sigma(\mathrm{Sym}_n)) \times \mathbb{R} \tag{4.3}$$

into strata that are compatible with the action of Sym_n, i.e., each stratum is sent to another stratum of same dimension by the action of Sym_n. This follows from Theorem 4.10 and the fact that $\Sigma(\mathrm{Sym}_n)$ is stratified and the map $(p|_{\mathbb{R}^n \setminus L}, q)$ of Theorem 4.9 is Sym_n-equivariant. The strata in Equation 4.3 are indexed by pairs of cosets $(\tau_1 P_1, \tau_2 P_2)$, where $\tau_1, \tau_2 \in \mathrm{Sym}_n$ and $P_1, P_2 \leq \mathrm{Sym}_n$ are parabolic subgroups[2]. The partial ordering on these pairs is given component-wise by reverse inclusion (cf. Equation 2.1).

It follows that the quotient $X = \mathrm{Sym}_n \backslash \mathbb{R}^{2n}$ is stratified over the quotient $\mathcal{P}$ of this poset by the action of Sym_n. More concretely, $\mathcal{P}$ can be described as follows: The elements of $\mathcal{P}$ are orbits of the form $\mathrm{Sym}_n \cdot (\tau_1 P_1, \tau_2 P_2)$, where $\tau_1, \tau_2 \in \mathrm{Sym}_n$ and $P_1, P_2 \leq \mathrm{Sym}_n$ are parabolic subgroups. The partial ordering is given by

$$\mathrm{Sym}_n \cdot (\tau_1 P_1, \tau_2 P_2) \leq \mathrm{Sym}_n \cdot (\tau_1' P_1', \tau_2' P_2')$$

if there is $\gamma \in \mathrm{Sym}_n$ such that

$$\tau_1 P_1 \supseteq \gamma \tau_1' P_1' \text{ and } \tau_2 P_2 \supseteq \gamma \tau_2' P_2'.$$

The poset $\mathcal{P}$ has a more explicit description in terms of another poset $\mathcal{Q}$, which has as elements "marked" double cosets of parabolic subgroups.

Definition 4.19. Let $\mathcal{Q}$ be the poset consisting of all triples $(P_1, P_1 \sigma P_2, P_2)$, where $\sigma \in \mathrm{Sym}_n$ and $P_1, P_2 \leq \mathrm{Sym}_n$ are parabolic subgroups and where

$$(P_1, P_1 \sigma P_2, P_2) \leq (P_1', P_1' \sigma P_2', P_2')$$

if and only if there is component-wise containment in the reverse direction,

$$P_1 \supseteq P_1',\ P_2 \supseteq P_2' \text{ and } P_1 \sigma P_2 \supseteq P_1' \sigma P_2'.$$

A very similar poset is also studied as a two-sided version of the Coxeter complex by Hultman [63] and Petersen [98]. We remark that $\mathcal{Q}$ is different from the poset of all double cosets of the form $P_1 \sigma P_2$: There can be $P_1 \neq P_1', P_2 \neq P_2'$ such that $P_1 \sigma P_2 = P_1' \sigma P_2'$ (see [98, Remark 4]).

[2] Note that, following Theorem 4.12, the points in $\mathrm{Conf}_n(\mathbb{R}) \times \mathrm{Conf}_n(\mathbb{R}) \subset \mathbb{R}^n \times \mathbb{R}^n$ are exactly the ones that belong to the top-dimensional strata. Similarly, the points of $L \times L \subset \mathbb{R}^n \times \mathbb{R}^n$ belong to the lowest dimensional strata, corresponding to the cone points in Equation 4.3.

Lemma 4.20. *The map*

$$\begin{gathered}\phi : \mathcal{P} \to \mathcal{Q}\\ \mathrm{Sym}_n \cdot (\tau_1 P_1, \tau_2 P_2) \mapsto (P_1, P_1 \tau_1^{-1} \tau_2 P_2, P_2)\end{gathered}$$

is an isomorphism of posets.

Proof. To see that ϕ is a bijection of the underlying sets, consider the following map:

$$\begin{gathered}\psi : \mathcal{Q} \to \mathcal{P}\\ (P_1, P_1 \sigma P_2, P_2) \mapsto \mathrm{Sym}_n \cdot (P_1, \sigma P_2).\end{gathered}$$

It is easy to verify that ϕ and ψ are independent of the choices of representatives and are inverse to one another. That ϕ is indeed a map of posets, i.e., that it preserves the partial ordering, follows from elementary manipulations of cosets. □

Theorem 4.21. *The set $\mathcal{B}_n$ of barcodes with n bars is stratified over the poset $\mathcal{Q}$. The lowest dimensional stratum containing the barcode B is the stratum corresponding to $(P_b^B, D_B, P_d^B) \in \mathcal{Q}$. It is of the form (using the notation of Theorem 2.23 and Theorem 4.16)*

$$\mathcal{B}_n^{(P_b^B, D_B, P_d^B)} = \big(\mathrm{Sym}_n \cdot (\mathrm{cone}(\tau_b P_b^B) \times \mathbb{R} \times \mathrm{cone}(\tau_d P_d^B) \times \mathbb{R})\big) \cap Y,$$

where $\mathrm{cone}(\tau P) = \tau P \times [0, \infty)/(x, 0) \sim (y, 0)$.

Proof. Recall that $\mathcal{B}_n \cong Y$ is a subset of $X = \mathrm{Sym}_n \backslash \mathbb{R}^{2n}$ (Theorem 4.14). As observed above, X is stratified over the poset $\mathcal{P}$ and, by Theorem 4.20, this poset is isomorphic to $\mathcal{Q}$. It follows that $\mathcal{B}_n$ is also stratified over $\mathcal{Q}$. The strata are obtained by taking the intersection with Y.

This stratification is induced by the simplicial structure of the Coxeter complexes in

$$X \cong \mathrm{Sym}_n \backslash \big(\mathrm{cone}(\Sigma(\mathrm{Sym}_n)) \times \mathbb{R} \times \mathrm{cone}(\Sigma(\mathrm{Sym}_n)) \times \mathbb{R}\big).$$

Hence, the strata that contain a barcode $B \in \mathcal{B}_n$ depend only on the coordinate $\mathrm{Sym}_n \cdot (b_\theta, d_\theta) \in \mathrm{Sym}_n \backslash \Sigma(\mathrm{Sym}_n) \times \Sigma(\mathrm{Sym}_n)$ that B determines by Theorem 4.15. As explained in Theorem 4.12, the associated points b_θ, $d_\theta \in \Sigma(\mathrm{Sym}_n)$ lie in the interior of the simplices $\tau_b P_b^B$, $\tau_d P_d^B$. Hence, the lowest dimensional stratum that contains B corresponds to the Sym_n-orbit of $(\tau_b P_b^B, \tau_d P_d^B)$. □

Let B be a strict barcode, that is, $b_i \neq b_j$ and $d_i \neq d_j$ for $i \neq j$. Then B is contained in the top-dimensional stratum

$$\mathcal{B}_n^{(\{\mathrm{id}\}, \{\mathrm{id}\}\tau_b^{-1}\tau_d\{\mathrm{id}\}, \{\mathrm{id}\})} = (\mathrm{Sym}_n \cdot (\mathrm{cone}(\tau_b \{\mathrm{id}\}) \times \mathbb{R} \times \mathrm{cone}(\tau_d \{\mathrm{id}\}) \times \mathbb{R})) \cap Y.$$

Changing the representative of the Sym_n-orbit, this can be rewritten as

$$\mathcal{B}_n^{(\{\mathrm{id}\},\{\sigma_B\},\{\mathrm{id}\})} = (\mathrm{Sym}_n \cdot (\mathrm{cone}(\{\mathrm{id}\}) \times \mathbb{R} \times \mathrm{cone}(\sigma_B \{\mathrm{id}\}) \times \mathbb{R})) \cap Y,$$

where $\sigma_B = \tau_b^{-1}\tau_d$ is the permutation associated to B as in Theorem 3.22. In particular, the strata containing strict barcodes are in one-to-one correspondence with the elements of Sym_n.

When one considers the cone and real line parameters in the stratification of Theorem 4.21, one obtains regions that are determined by the averages and standard deviations of Theorem 4.15 and by parabolic subgroups.

Corollary 4.22. *The Coxeter coordinates of Theorem 4.15 decompose the space $\mathcal{B}_n$ of barcodes with n bars into disjoint regions. The region containing the barcode $B = \{(b_i, d_i)\}_{i\in\{1,\ldots,n\}} \in \mathcal{B}_n$ is defined as the set of all barcodes B' such that:*

1. *its average birth time is the same as that of B, i.e., $\bar{b'} = \bar{b}$;*
2. *its average death time is the same as that of B, i.e., $\bar{d'} = \bar{d}$;*
3. *its birth standard deviation is the same as that of B, i.e., $\|v_{b'}\| = \|v_b\|$;*
4. *its death standard deviation is the same as that of B, i.e., $\|v_{d'}\| = \|v_d\|$;*
5. *$P_b^{B'} = P_b^B$, $P_d^{B'} = P_d^B$ and $D_B = D_{B'}$.*

For strict barcodes, the information of the last Item 5 is equivalent to specifying σ_B, the permutation associated to barcodes in Theorem 3.22.

4.2.4 A metric on $\mathcal{B}_n$

In this section, we explain how the description of $\mathcal{B}_n$ given in Section 4.2.2 with $\mathbb{R}^n$ equipped with the l^∞-norm gives rise to a naturally defined metric $\tilde{d}_B$ on $\mathcal{B}_n$ that is closely related to the bottleneck distance. Similarly, the l^2-norm on $\mathbb{R}^n$ leads to a modified Wasserstein distance $\tilde{d}_W$ on $\mathcal{B}_n$.

To describe $\tilde{d}_B$, we equip $\mathbb{R}^{2n}$ with the metric d_∞ induced by the l^∞-norm. This metric induces a map $X \times X \to \mathbb{R}$ on the quotient by taking the minimum value over all representatives of the corresponding equivalence classes:

$$\begin{aligned} d : X \times X &\to \mathbb{R} \\ \big([x,y],[x',y']\big) &\mapsto \min_{\substack{(\tilde{x},\tilde{y})\in[x,y],\\ (\tilde{x}',\tilde{y}')\in[x',y']}} d_\infty(\,(\tilde{x},\tilde{y}),(\tilde{x}',\tilde{y}')\,). \end{aligned} \tag{4.4}$$

We will show that this map restricted to Y agrees with a modified version of the bottleneck distance.

Definition 4.23. Let $B = \{(b_i, d_i)\}_{i\in\{1,\ldots,n\}}$ and $B' = \{(b'_i, d'_i)\}_{i\in\{1,\ldots,n\}}$ be two barcodes in $\mathcal{B}_n$. The *modified bottleneck distance* between B and B' is

$$\tilde{d}_B(B, B') \coloneqq \min_{\gamma\in\mathrm{Sym}_n} \max_{i\in\{1,\ldots,n\}} \|(b_i, d_i) - (b'_{\gamma(i)}, d'_{\gamma(i)})\|_\infty.$$

where $\|\cdot\|_\infty$ is the l^∞-norm on $\mathbb{R}^2$.

Note that the difference between the modified bottleneck distance and the original bottleneck distance as defined in Theorem 3.25 is that for the modified version, one is not allowed to match points of the barcodes to the diagonal Δ (see Figure 4.7). Furthermore, $\tilde{d}_B(B, B')$ is well-defined only if both B and B' contain the same number of bars, i.e., if they are both elements of the same $\mathcal{B}_n$. This is not necessary for the definition of the regular bottleneck distance, cf. Theorem 4.25.

Proposition 4.24. *The map d defines a metric on Y with respect to which $\phi : (\mathcal{B}_n, \tilde{d}_B) \longrightarrow (Y, d)$ is an isometry.*

Proof. As observed before in Theorem 4.14, ϕ maps $\mathcal{B}_n$ bijectively onto Y. Hence, it is sufficient to show that for arbitrary barcodes B and B',

$$\tilde{d}_B(B, B') = d(\phi(B), \phi(B')).$$

This follows from simply spelling out the definitions. For points (x, y) and (x', y') in $\mathbb{R}^n \times \mathbb{R}^n$,

$$\begin{aligned} d_\infty((x, y), (x', y')) &= \max\left\{|x_1 - x'_1|, \ldots, |x_n - x'_n|, |y_1 - y'_1|, \ldots, |y_n - y'_n|\right\} \\ &= \max_{i\in\{1,\ldots,n\}} \max\left\{|x_i - x'_i|, |y_i - y'_i|\right\} \\ &= \max_{i\in\{1,\ldots,n\}} \|(x_i, y_i) - (x'_i, y'_i)\|_\infty, \end{aligned}$$

where $\|\cdot\|_\infty$ is the l^∞-norm on $\mathbb{R}^2$. Combining this with the definition of d on X (see Equation 4.4), we obtain

$$\begin{aligned} d(\phi(B), \phi(B')) &= \min_{\gamma\in\mathrm{Sym}_n} d_\infty(\,\phi(B), \gamma \cdot \phi(B')\,) \\ &= \min_{\gamma\in\mathrm{Sym}_n} \max_{i\in\{1,\ldots,n\}} \|(b_i, d_i) - (b'_{\gamma^{-1}(i)}, y'_{\gamma^{-1}(i)})\|_\infty. \end{aligned}$$

This is the same as the modified bottleneck distance of Theorem 4.23. □

Similarly, starting with $\mathbb{R}^{2n}$ equipped with the l^2-norm, one can establish an isometry between Y and $\mathcal{B}_n$ equipped with a modified Wasserstein distance instead.

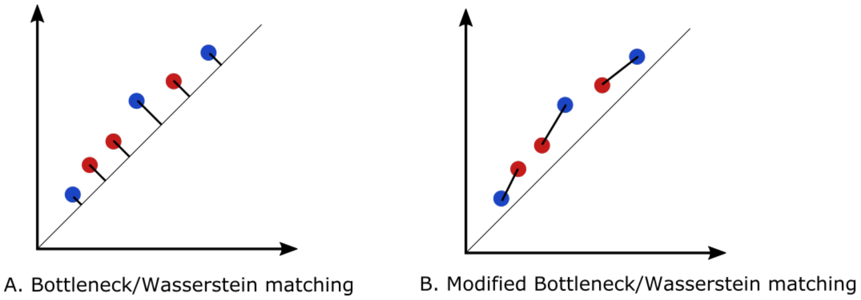

A. Bottleneck/Wasserstein matching
B. Modified Bottleneck/Wasserstein matching

Figure 4.7: Two barcodes (red and blue) represented as persistence diagrams in $\mathbb{R}^2$. A. The matching that minimises the bottleneck or Wasserstein distance matches all the bars to the diagonal, as they are all very close to it. B. If bars are not allowed to be matched with the diagonal, the matching that minimises $\|(b_i, d_i) - (b'_{\gamma(i)}, y'_{\gamma(i)})\|_\infty$ for the bottleneck or $\sum_i \|(b_i, d_i) - (b'_{\gamma(i)}, y'_{\gamma(i)})\|_2$ respectively for the Wasserstein is different.

Remark 4.25. Forgetting about the diagonal as done above opens the door to defining new metrics on barcodes by considering distances on $\mathbb{R}^n \times \mathbb{R}^n$ and then taking the quotient as was done in this section. It could potentially be extended to barcodes with different number of bars. One could for instance imagine a map that forces matchings between as many bars as possible and then adds a positive weight equal to their distance to the diagonal to the unmatched bars if there are any. This is different from the bottleneck distance (or Wasserstein distance), which allows as many matchings as needed with the diagonal, see Figure 4.7. When using barcodes to study data, bars close to the diagonal are usually considered as related to noise. However, there are cases where all the bars matter, for instance when the barcode is the one of a merge tree [32, 69]. In such a case, a new metric that does not take the diagonal into account could turn out useful. We leave this for future work.

Chapter 5

Inverse Problem: From Trees to Barcodes and Back Again

As described in Section 3.3, every merge tree has an associated barcode. It is natural to ask whether the map from merge trees to barcodes determined by the Elder rule is injective, but it is not hard to see that it is not. A somewhat more surprising result, proven independently in [31] and [69], is that the failure of injectivity of the Elder rule map can be quantified combinatorially for generic barcodes. In this chapter, we study the inverse problem of trees and barcodes from a combinatorial point of view, using the identification of barcodes with elements of the symmetric group studied in Chapter 4. This chapter is based on joint work with J. Curry, J. DeSha, K. Hess, L. Kanari, B. Mallery, [32, 69].

We summarize the notation for this chapter in the box below.

Strict barcode: a barcode $B = \{(b_i, d_i)\}_{i\in\{0,\dots,n\}}$ such that the essential bar (b_0, d_0) contains all the others bars and $b_i \neq b_i, d_i \neq d_j$ if $i \neq j$. The birth values are assumed to be ordered $b_0 < \dots < b_n$.
Births: they correspond to the termination points of the branches, i.e., the tips of the branches.
Deaths: they correspond to the bifurcation points of the branches, i.e., the internal nodes.
Permutation: the permutation associated to a strict barcode B with ordered birth times is computed via $\sigma_B(i) = \#\{j < i \mid d_i \leq d_j\}$. The essential bar (b_0, d_0) is not considered.
The space $\mathcal{B}_n$: it consists of all barcodes with $n+1$ bars, including the essential one (b_0, d_0).

5.1 The Tree Realization Number

5.1.1 Realizing Barcodes as Trees

We say that a merge tree (T, h) *realizes* a barcode B if the barcode of (T, h) is B.

Definition 5.1. The *tree realization number* (TRN) $R(B)$ of a strict barcode B is the number of combinatorial trees T admitting a height function h such that (T, h) realizes B.

We show later in this chapter that this number depends only on the combinatorial type of barcode, i.e., the permutation associated to it. For the remainder of this chapter, we assume, for convenience, that the bars are ordered by birth times.

Example 5.2. Examples of tree-realizations are provided in Figure 5.1B. We encode the combinatorial structure of the tree, i.e., how the branches may be attached to each other, in an adjacency matrix in which the (i, j) coefficient is non-zero if the Elder Rule allows bar i to be connected to bar j. For example, in Figure 5.1A, bars $1 - 3$ may all be connected to the black bar 0, thus the coefficients $(0, 1), (0, 2), (0, 3)$ are all non-zero in the corresponding adjacency matrix. Note that in each realization only a subset of these possible attachments is actually made (Figure 5.1B), since each branch can be attached to only one other branch. The connectivity diagram (bottom of Figure 5.1A) provides another representation of the pairs of branches that may be connected, in agreement with the Elder Rule. The arrow on an edge in the diagram indicates the direction of the connection. In this example, there are arrows from 0 towards $1, 2$, and 3, from 1 to 2 and 3, and from 2 to 3.

We provide a formula for $\mathrm{R}(B)$ in terms of the *index* of each bar (b_i, d_i), i.e., the number of bars that include the i-th bar (b_i, d_i) strictly:

$$\mathrm{index}_i(B) = \#\{j \mid b_j < b_i < d_i < d_j\} = \#\{j < i \mid d_i < d_j\}.$$

A version of this formula was established independently by Curry in [31].

Proposition 5.3. *Let $B = \{(b_i, d_i)\}_{i\in\{0,\dots,n\}}$ be a strict barcode. Its tree-realization number is equal to the product of the indices of its bars, i.e.,*

$$\mathrm{R}(B) = \prod_{1\leq i\leq n} \mathrm{index}_i(B).$$

Proof. Because of the Elder Rule, one branch can be attached to another only if its corresponding bar is included in the other bar. This simple observation enables us to prove the lemma by a straightforward recursion on the number of bars. □

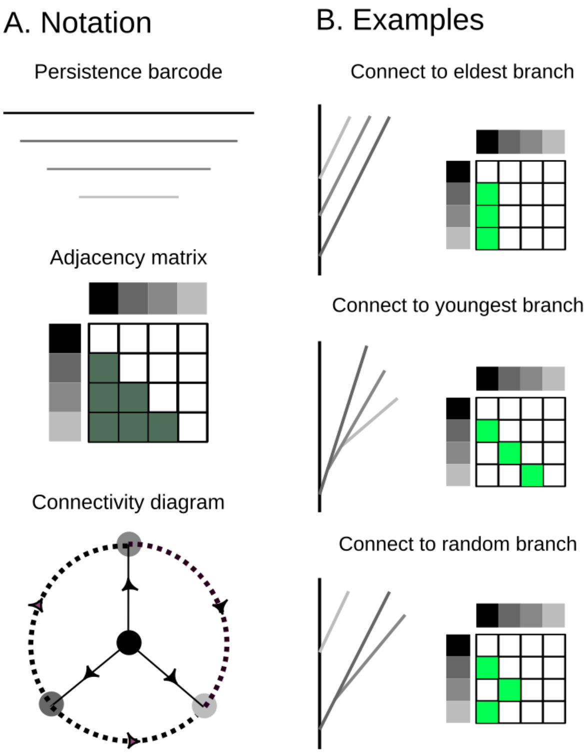


Figure 5.1: A strict barcode, whose bars are ordered according to birth times (greyscale), defines a unique ordering of death times. This ordering and the Elder Rule constrain the possible combinatorial types of trees that can be realized from this barcode. A. The notation that derived from a barcode that corresponds to an adjacency matrix of possible connectivities. Equivalently the possible connectivities are presented in the connectivity diagram. B. Examples of possible tree realizations.

In particular, the maximum tree-realization number for a strict barcode with $n+1$ bars is $n!$, in the specific case where $d_n < ... < d_1 < d_0$. We call this case the *Russian doll barcode.*

Example 5.4. Consider the strict barcode $B = \Big\{(0,10),(1,8),(2,7),(3,6),(4,5)\Big\}$. According to the formula in Proposition 5.3,

$$\prod_{1\leq i\leq n} \text{index}_i(B) = 1\cdot 2\cdot 3\cdot 4 = 4!.$$

Based on Proposition 5.3, there is a recursive process to build all trees realizing a given barcode. As we formalize in Section 5.2.1, it depends only on the permutation type of barcode. Figure 5.2 shows a graphical representation of this process to obtain all trees that have the same barcode up to combinatorial equivalence class.

We provide a brief sketch here for the sake of intuition. Start by setting $T_0 = I_0 = (b_0, d_0)$, the trunk of the tree corresponding to the essential bar. Since the merge tree T is connected, we can recursively attach bars by death time, first to T_0 and then in the j^{th} step to T_j to get T_{j+1}, according to the Elder rule. Each possible choice of attachment then gives a particular merge tree isomorphism class.

Note that the tree-realization number does **not** satisfy

$$\mathrm{R}(B) = \mathrm{R}(B') \implies B \sim B'$$

in general, i.e., the tree-realization number is not a complete invariant of the barcode equivalence relation. For instance, the barcodes corresponding to permutations [231] and [312] in Figure 5.2 both have $\mathrm{R}(B) = 2$ but have different permutation types. The inverse clearly does hold, however:

$$\mathrm{R}(B) \neq \mathrm{R}(B') \implies B \not\sim B',$$

enabling us to detect non-equivalence of barcodes.

5.1.2 Combinatorics of the TRN

In Chapter 6, we use the TRN as an invariant to detect whether the type of barcodes has changed. For the specific cases studied in Chapter 6, it is usually true that if barcodes have same TRN, then they are equivalent. Therefore, the TRN is useful to detect equivalence of barcodes in these special cases. In particular, if a new bar is added to a barcode or two deaths are transposed and no other changes take place, then the tree-realization number does change. Lemma 5.5 enables us to quantify how adding a new bar changes tree-realization number.

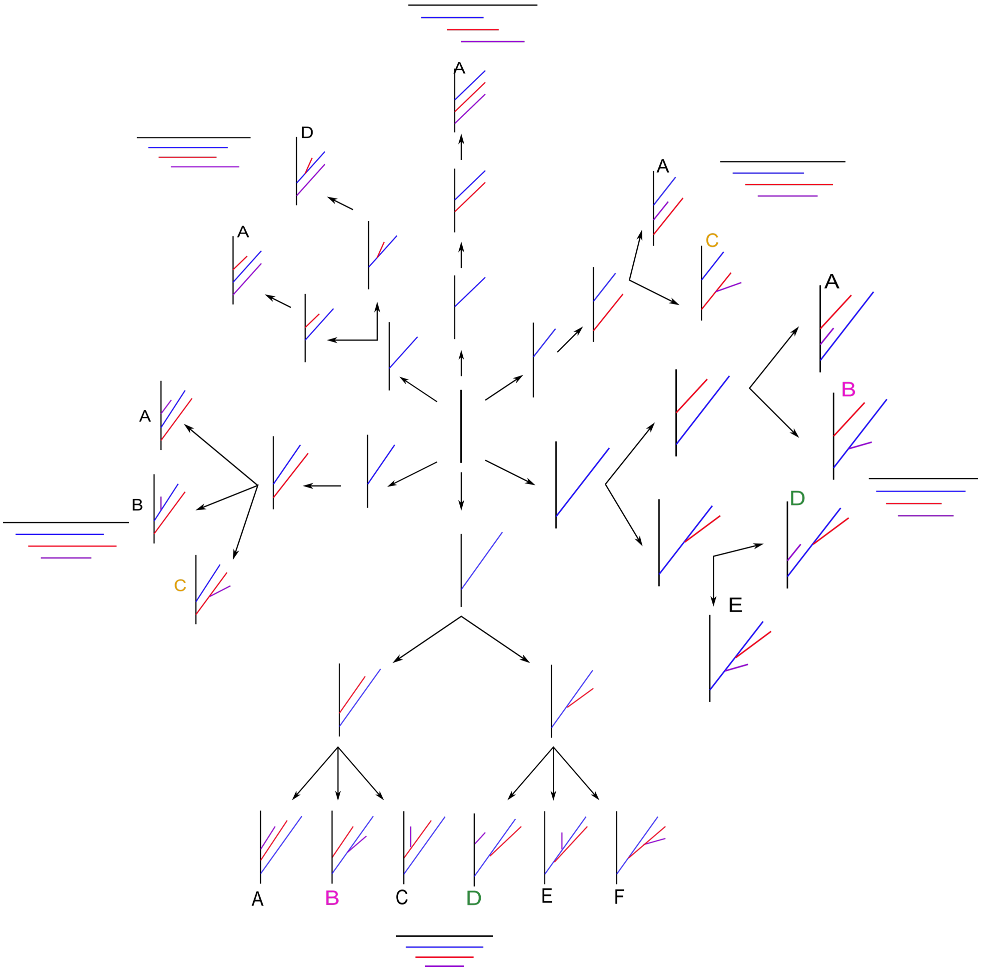


Figure 5.2: Recursive construction of all trees realising barcodes with three non-essential bars. At each bifurcation, the number of new branches corresponds to the index of the branch that is added. Each time we add a new branch, we multiply the number of possibilities by its index, illustrating the result of Proposition 5.3. The colored indices show the pairs of different combinatorial type of merge trees which have the same combinatorial type of phylogenetic trees.

Lemma 5.5. *Let* $B = \{(b_i, d_i)\}_{i \in \{0,\dots,n\}}$ *and* $B' = B \cup \{(b_{n+1}, d_{n+1})\}$*, where* $b_{n+1} > b_i$ *for all* $0 \leq i \leq n$*, be strict barcodes. If* $d_{i_1} > \dots d_{i_{k-1}} > d_{n+1} > d_{i_k} > \dots d_{i_n}$*, then*

$$\mathrm{R}(B') = \mathrm{R}(B) \cdot k.$$

Proof. The condition on d_{n+1} implies that the new bar (b_{n+1}, d_{n+1}) is included in exactly k other bars, so its index is k. □

Example 5.6. Let B be a barcode with four bars such that $b_0 < b_1 < b_2 < b_3$ and $d_0 > d_2 > d_1 > d_3$, i.e., its equivalence class is [312]. It is easy to see that $\mathrm{R}(B) = 3$ (see Figure 5.3). If we add a new bar (b_4, d_4) such that $d_1 > d_4 > d_3$, the equivalence class of the new barcode B' is [3412], and bar (b_4, d_4) is included in (b_0, d_0), (b_2, d_2) and (b_1, d_1), but not in (b_3, d_3) because $d_4 > d_3$. Therefore, its index is 3, whence $\mathrm{R}(B') = 3 \cdot 3 = 9$.

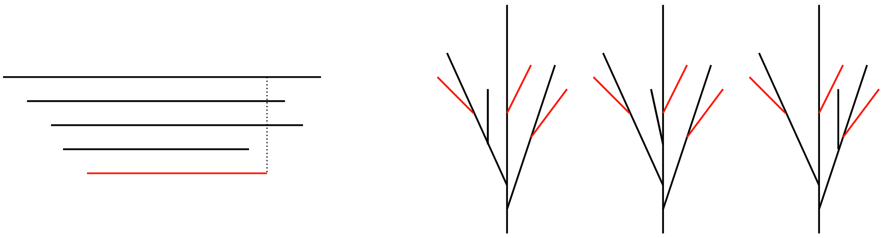


Figure 5.3: A barcode B in equivalence class [312] is shown in black. There are three possible combinatorial equivalence classes of trees whose TMD barcode is B, also represented in black. After adding the extra bar in red, we obtain a new barcode B', in the equivalence class [3412]. In a tree-realization of B', the branch corresponding to the red bar can be attached to any of the branches corresponding to the 0th, 1st, and 2nd bars, represented on the trees by the red branches. This leads to nine possible combinatorial equivalence classes of trees for the barcode [2412].

We can also apply Proposition 5.3 to determining how switching the order of two consecutive deaths in the barcodes affects the tree realization number.

Proposition 5.7. *Let* $B = \{(b_i, d_i)\}_{i \in \{0,\dots,n\}}$ *be a strict barcode in the equivalence class* $[i_1 \dots i_n]$*, that is,* $d_{i_1} < d_{i_2} < \dots < d_{i_n}$*. Let* $B' = \{(b'_i, d'_i)\}_{i \in \{0,\dots,n\}}$ *be a new barcode obtained by permuting the deaths* d_{i_k} *and* $d_{i_{k+1}}$*, i.e.,* $b_i = b'_i$ *for all* i *and* $d_i = d'_i$ *for all* $i \neq i_k, i_{k+1}$*, while* $d_{i_k} = d'_{i_{k+1}}$ *and* $d_{i_{k+1}} = d'_{i_k}$*.*

1. *If* $i_k < i_{k+1}$*, then* $\mathrm{index}_{i_{k+1}}(B') = \mathrm{index}_{i_{k+1}}(B) + 1$*, and*

$$\mathrm{R}(B') = \frac{\mathrm{R}(B)(\mathrm{index}_{i_{k+1}}(B) + 1)}{\mathrm{index}_{i_{k+1}}(B)}.$$

2. If $i_k > i_{k+1}$*, then* $\mathrm{index}_{i_{k+1}}(B') = \mathrm{index}_{i_{k+1}}(B) - 1$*, and*

$$\mathrm{R}(B') = \frac{\mathrm{R}(B)(\mathrm{index}_{i_{k+1}}(B) - 1)}{\mathrm{index}_{i_{k+1}}(B)}.$$

Proof. It is enough to prove (1), since (2) then follows by switching the roles of B and B'.

If $i_k < i_{k+1}$, then $b_{i_k} < b_{i_{k+1}}$. Since B is in the equivalence class $[i_1...i_n]$, $d_{i_k} < d_{i_{k+1}}$, whence $(b_{i_{k+1}}, d_{i_{k+1}}) \not\subset (b_{i_k}, d_{i_k})$. On the other hand, $(b'_{i_{k+1}}, d'_{i_{k+1}}) \subset (b'_{i_k}, d'_{i_k})$, but otherwise respects all of the same inclusion relations as $(b_{i_{k+1}}, d_{i_{k+1}})$, so that

$$\mathrm{index}_{i_{k+1}}(B') = \mathrm{index}_{i_{k+1}}(B) + 1,$$

as desired. Moreover, $(b'_{i_k}, d'_{i_k}) \not\subset (b'_{i_{k+1}}, d'_{i_{k+1}})$, so (b'_{i_k}, d'_{i_k}) respects exactly the same inclusion relations as (b_{i_k}, d_{i_k}), i.e.,

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

Because no other bars are affected when passing from B to B', we can conclude. □

Example 5.8. In Figure 3.8, we show all the possible death-transpositions in a strict barcode with five bars with their TRN indicated next to each permutation type. As an example, take the two barcodes of Figure 5.4. Barcode B is in the equivalence class [4312], so the barcode satisfies $d_4 < d_3 < d_1 < d_2$. The index of (b_4, d_4) is 4, because it is included in all the other bars. Permuting d_3 and d_4 leads to a barcode B' in the equivalence class [3412]. The index of the last bar is now 3 because it is no longer included in the third bar.

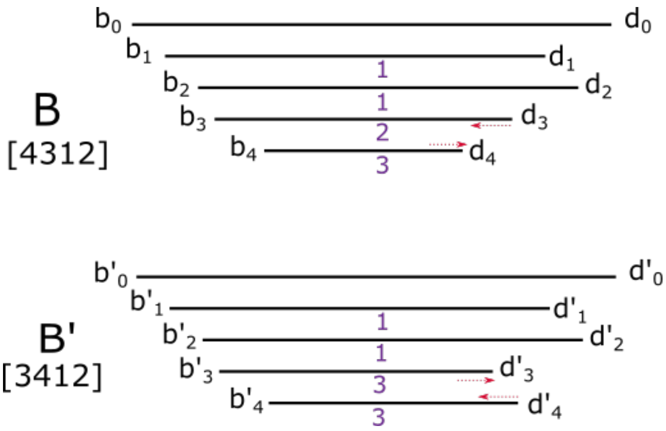


Figure 5.4: Illustration of Proposition 5.7. Two barcodes B and B' that have the same birth and death values apart from d_3 and d_4, which are switched to d'_4 and d'_3 in B'. The index of each bar is indicated in purple below.

The reader might have noticed that for the last examples, we used barcodes up to combinatorial equivalence class. The reason for this slight abuse is that the TRN depends only on the type of permutation of the barcode, which we prove in the next section.

We illustrate the results of Lemma 5.5 and Proposition 5.7 on the Cayley graphs of Sym_3 and Sym_4 in Figures 5.5B and 3.8.

Figure 5.5 shows the Cayley graph of Sym_3 generated by the permutations (12) and (23) and the corresponding equivalence classes of barcodes. The vertices of the graph correspond to the permutations in the symmetric group and their corresponding barcode types, and the edges between them to the transpositions transforming one permutation into another. The number next to each bar is its index. The trees that return a given barcode are sketched next to each vertex of the Cayley graph of Sym_3.

5.2 Combinatorial Perspective

In this section, we finally formalize the inverse problem of trees and barcodes from a combinatorial point of view: the TRN depends only on the permutation type of barcodes. The commuting diagram of Figure 3.10 makes even more sense in this setting. To understand how many combinatorial trees have the same barcodes, one can look only at the bottom part of the diagram, the combinatorial Elder rule between combinatorial merge trees and permutations. This suffices to determine the TRN for the whole equivalence class of a barcode.

We then go on to study more combinatorial properties of the TRN: since it depends only on the combinatorial type of barcode, i.e., the permutation type, we show that it can be computed from a commonly studied object for permutations: the left inversion vector. It follows easily that the TRN preserves the Bruhat order. Finally, we use this combinatorial knowledge about the TRN to study the sum of the realization numbers, i.e., the number of combinatorial merge trees. We show that this number is different from the number of phylogenetic trees, as foreseen in Chapter 3.

5.2.1 The Combinatorial Inverse Problem

We show here that the study of the trees that have the same barcodes reduces to a combinatorial study of the type of barcodes: the TRN depends only on the permutation type of barcodes.

For any strict barcode B, let $\mathcal{T}(B)$ denote the set of combinatorial equivalence classes of tree-realizations of B, i.e.,

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

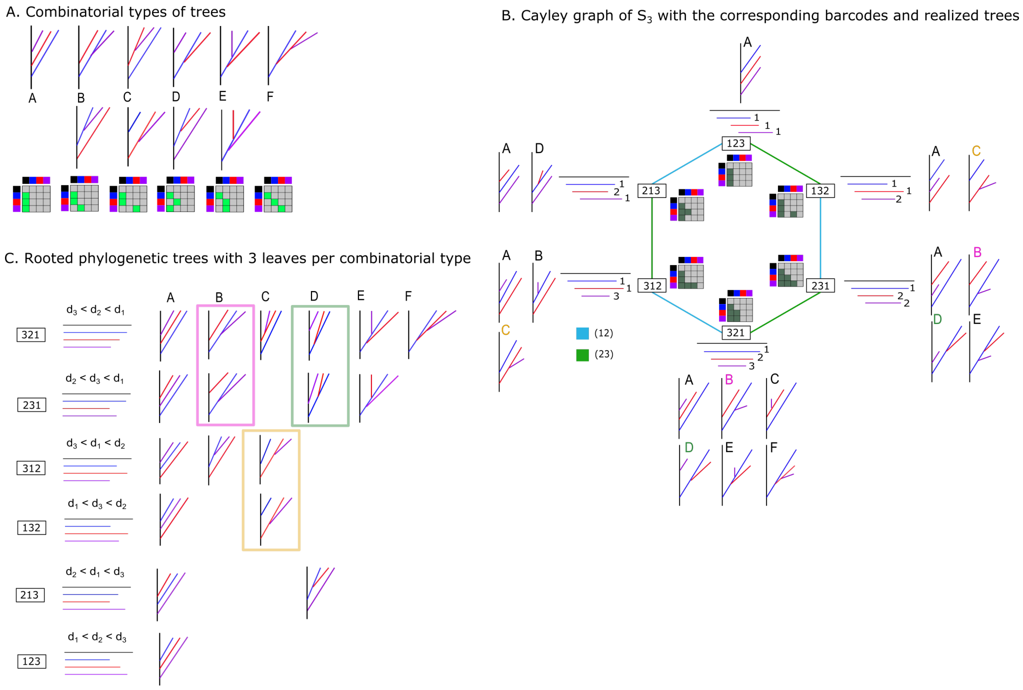

A. Combinatorial types of trees
A
B
C
D
E
F
B. Cayley graph of S3 with the corresponding barcodes and realized trees
A
123
213
132
312
231
321
A
D
A
C
A
B
C
A
B
D
E
(12)
(23)
A
B
C
D
E
F
C. Rooted phylogenetic trees with 3 leaves per combinatorial type
A
B
C
D
E
F
321
d3 < d2 < d1
231
d2 < d3 < d1
312
d3 < d1 < d2
132
d1 < d3 < d2
213
d2 < d1 < d3
123
d1 < d2 < d3

Figure 5.5: A. Combinatorial types of rooted trees with three leaves and the corresponding adjacency matrices. B. Cayley graph generated by the two adjacent transpositions of Sym_3 and the corresponding barcodes, together with all the combinatorial types of trees that realize a barcode. Colored letters correspond to different types of merge trees that are the same as phylogenetic trees (indistinguishable trees), illustrating the result of Section 5.2.5. C. Rooted phylogenetic trees with three leaves. We represent these phylogenetic trees organized by the cominatorial types of barcodes they would have if they had death labels as well. The three pairs of trees within colored squares correspond to the indistinguishable trees defined in Section 5.2.5: the internal nodes are incomparable, so they can have two different death values that lead to different merge trees. In phylogenetic trees, the label order does matter: for instance, in the first column, all the trees are of the same combinatorial type A but correspond to different phylogenetic trees. To go from the space of phylogenetic trees to the space of combinatorial trees, one forgets the labels and considers the adjacencies only, see Figure 3.6.

We can characterize the equivalence relation on strict barcodes in terms of $\mathcal{T}(B)$.

Lemma 5.9. *If B and B' are two strict barcodes with the same number of bars, then*

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

Proof. The order of the deaths in a strict barcode B completely determines the set of combinatorial equivalence classes of its possible tree realizations.

Indeed, the two pairs of bars in Figure 5.6(2) lead to the same adjacency possibilities for their respective branches. Only move (1) in Figure 5.6, corresponding to switching the order of the deaths of the two bars, modifies the permutation equivalence class of the barcode, hence also the set of trees that return the given barcode.

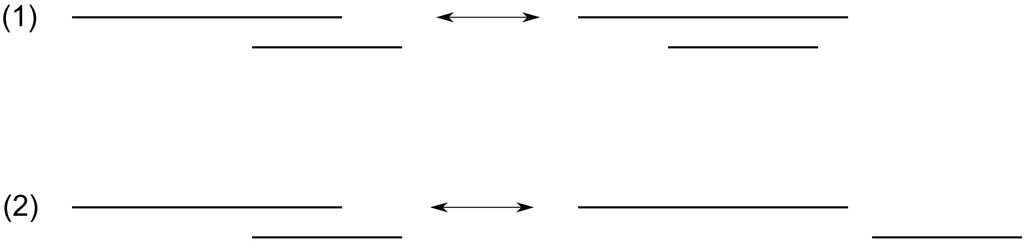


Figure 5.6: The two possible moves that respect the condition of a realisable barcode. Move (1) modifies the barcode's ordering, whereas move (2) does not change the order of the deaths.

□

5.2.2 The TRN and the Left Inversion Vector

Inspection of the definition of the index of a bar (b_i, d_i) in a barcode B reveals that it is given by the number of bars born before b_i and that die after d_i. Thinking in terms of the permutation associated to a barcode, this index counts the number of inversions of birth-mapping-to-death order. More precisely, for a permutation σ of $\{1, \dots, n\}$ if $i < j$ and $\sigma(i) > \sigma(j)$, then either the pair of places (i, j) or the pair of elements $(\sigma(i), \sigma(j))$ is called an *inversion* of σ—the usual order $i < j$ has been "upset" or inverted here. We now modify the usual notion of an inversion vector so that it is defined for strict barcodes and makes our theorem statements as tidy as possible.

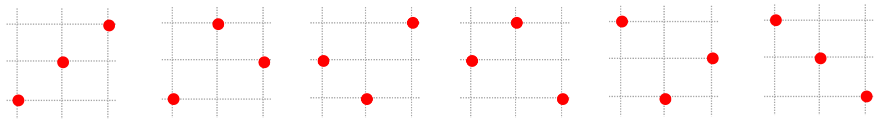
Definition 5.10. Let $B = (b_0, d_0) \cup \{[b_i, d_i)\}_{i \in \{1,\dots,n\}}$ be a strict barcode with $b_i < b_j$ for $i < j$. The *left inversion vector* of B is the n-vector $l(B)$ whose i-th coordinate is

$$l_i(B) := \#\{j \leq i \mid d_j \geq d_i\} = \mathrm{index}_i(B).$$

We note that, for this formula, the index $j = 0$ is used for computation although it is not given a position in the n-vector $l(B)$, since the vector would have length $n + 1$. When we calculate the left inversion vector of a permutation σ associated to a barcode, we use the slightly modified definition

$$l_i(\sigma) := \#\{j \leq i \mid \sigma(j) \geq \sigma(i)\}$$

in order to make sure that $l(\sigma) = l(B)$.



permutation	id	(23)	(12)	(123)	(132)	(13)
left inversion vector	(1,1,1)	(1,1,2)	(1,2,1)	(1,1,3)	(1,2,2)	(1,2,3)
tree realization number	1	2	2	3	4	6

Figure 5.7: Persistence diagrams associated to the six elements of Sym_3, along with their inversion vectors and tree realization numbers.

Example 5.11. One can easily compute the left inversion vector of the following barcode:

$$B = \{(0, 10), (1, 7), (2, 6), (3, 5), (4, 8)\} \quad \Rightarrow \quad l(B) = (1, 2, 3, 1).$$

The permutation associated to this barcode is $\sigma = [3214]$ because the first bar to die corresponds to the third birth, the second to the second, the third to the first and the fourth to the fourth. Clearly, $l(\sigma) = (1, 2, 3, 1)$ as well.

Example 5.12. For the left inversion vectors associated to the six elements of Sym_3, along with their tree realization numbers, see Figure 5.7.

To define coordinates on the space of left inversion vectors, we use the the totally ordered sets

$$[k] := \{1 < 2 < \cdots < k\}$$

for k a positive natural number. It is easy to see that the left inversion vector construction establishes a bijective correspondence between Sym_n and the Cartesian product of sets of the above form, i.e., there is a bijection

$$l : \mathrm{Sym}_n \longrightarrow [1] \times [2] \times \cdots \times [n-1] \times [n] \qquad \text{where} \qquad \sigma \mapsto l(\sigma).$$

The next lemma, which is crucial for the rest of this section, follows immediately from this observation.

Lemma 5.13. *If $B = \{(b_i, d_i)\}_{i\in\{0,\dots,n\}}$ is a strict barcode, then*

$$R(B) = \prod_{i=1}^{n} l_i(\sigma(B)).$$

An immediate consequence of this lemma, which we already stated in the previous section, is that if B and B' are combinatorially equivalent barcodes, then their realization numbers are the same. It follows that the tree realization number induces a function on the symmetric group, i.e.,

$$R : \mathrm{Sym}_n \to \mathbb{N} : \sigma \mapsto \prod_{i=1}^{n} l_i(\sigma).$$

Before analyzing this function on the symmetric group, we identify some interesting properties of the set of barcodes under the combinatorial equivalence relation, to prepare our exploration of the combinatorics of the TRN in earnest in subsequent sections.

5.2.3 The TRN preserves the Bruhat Order

It is interesting to study both the tree realization number from a combinatorial point of view via the symmetric group and the symmetric group from a "barcode" point of view via the realization number. To the best of our knowledge, the product of the components of the left inversion vector is not a very commonly used statistic on symmetric groups, so we take this opportunity to study some of its properties.

Observe first that two adjacent permutations in the Cayley graph (i.e., two permutations that differ by left multiplication by one elementary transposition $\tau_i = (i, i+1)$) never have the same realization number. This follows easily from the definition. As a consequence, the realization number is locally injective, although it is not globally injective, since barcodes of type (12) and type (23) have the same TRN. In this section we extend this local injectivity observation, proving that the TRN defines an order-preserving map from the symmetric group to the natural numbers, when the symmetric group is equipped with the appropriate Bruhat order.

Example 5.14. In Sym_3 we note that $(123) > (23)$ under the left Bruhat order because $(123) = (12)(23)$, where we use cycle notation and where composition is read from right to left. In the left Bruhat order (123) and (12) are not comparable; see Figure 4.1.

The next lemma shows that the realization number increases with increasing left Bruhat order. We remark that this lemma can be viewed as a consequence of a classical result, which is mentioned in [43]: if $\sigma < \sigma'$, then the number of inversions in σ' is greater than the number of inversions in σ.

Lemma 5.15. *If $\sigma, \sigma' \in \mathrm{Sym}_n$ are such that $\sigma < \sigma'$ in the left Bruhat order, then $R(\sigma) < R(\sigma')$.*

Proof. Since $\sigma < \sigma'$, there exist $\tau_{i_1}, ..., \tau_{i_k} \in \mathcal{A}$ such that $\sigma' = \tau_{i_1} \cdots \tau_{i_k}\sigma$. If $k = 1$, then σ and σ' are adjacent on the Cayley graph, i.e., $\sigma' = \tau_i\sigma$ for some i. By assumption, the length of σ' is greater than that of σ.

Translating Proposition 5.7 into the language of permutations, we deduce that

$$R(\sigma') = \frac{R(\sigma)(l_{i+1}(\sigma) + 1)}{l_{i+1}(\sigma)} > R(\sigma).$$

The result now follows by induction on the number of transpositions τ_i. □

Example 5.16. One can see the Cayley graph of Sym_4 in Figure 3.8. Notice that two permutations σ, σ' satisfy $\sigma < \sigma'$ in the Bruhat order if and only if the shortest path from σ' to the identity contains the shortest path from σ to the identity. The realization number increases along such paths.

Remark 5.17. It is interesting to consider the TRN as a discrete Morse function [52] on the order complex of Sym_n. We note that the TRN has a unique max and min on Sym_n, which appear to be the only critical points, recovering the known result, e.g. [43], that the order complex of Sym_n is homotopy equivalent to a sphere.

5.2.4 The Sum of Realization Numbers and Chains in the Lattice of Partitions

Given that the tree realization number on the set of strict barcodes induces a function $R : \mathrm{Sym}_n \to \mathbb{N}$, it is natural to study the sum:

$$\sum_{\sigma \in \mathrm{Sym}_n} R(\sigma).$$

This sum is equal to the number of combinatorial classes of merge trees (Definition 3.9) and provides another quantitative characterization of the difference between merge trees and phylogenetic trees, which is explored further in the next section.

The sum of TRNs also connects this work with a classical object of study in algebraic combinatorics: each combinatorial equivalence class of merge trees corresponds to a maximal chain in the lattice of partitions, ordered by refinement. For topologists this should make intuitive sense: as two connected components merge this coarsens the partition of a sublevel set into connected components. Enumerating these components leads naturally to the study of the partitions of the set of $\{0, 1, \ldots, n\}$. We study this correspondence in more details in Section 5.2.6.

We start now by showing that this sum counts combinatorial equivalence classes of merge trees, but first prove a preparatory lemma.

Lemma 5.18. *If (T, h) and (T', h') are combinatorially equivalent merge trees with associated barcodes B and B', then the straight-line path $\overline{BB'}$ from B to B' lifts to a continuous path (with respect to the interleaving distance) connecting T and T'.*

Proof. Lemma 3.32 guarantees that the barcodes B and B' associated to T and T' have the same permutation type, so that the straight-line path $\overline{BB'}$ of Remark 4.4 does indeed exist, and every point on the path is a barcode of that permutation type by Lemma 4.3. We now apply the Elder Rule to construct a one-parameter family of merge trees

$$[0,1] \to \mathcal{MT}_n : t \mapsto (T^t, h^t)$$

that lifts the path $\overline{BB'}$.

Since (T, h) and (T', h') are combinatorially equivalent, the trees T and T' are isomorphic as graphs. Without loss of generality, we can suppose that $T = T'$.

To define our one-parameter family of merge trees, we set $T^t = T$ for all $t \in [0, 1]$ and specify the height function $h^t : V(T) \to \mathbb{R}$ as follows. We have no choice but to set $h^t(r) = \infty$, where r is the root, so it remains only to define h^t on the non-root nodes.

If v_i is the i-th leaf node by birth order in T, and therefore corresponds to the i-th bar of B^t, then the $h^t(v_i)$ is chosen to be the birth time of this bar, i.e.,

$$h^t(v_i) = b_i(1 - t) + b'_i t.$$

Similarly, if w_i is the internal node corresponding to the i-th bar in B^t, then $h^t(w_i)$ is chosen to be the death time of this bar, i.e.,

$$h^t(w_i) = d_i(1 - t) + d'_i t.$$

By construction, the barcode associated to (T, h^t) is clearly B^t.

It was shown in [85] (Theorem 2.2) that the interleaving distance between two merge trees in bounded by the maximal difference between the two height functions. Since $T^{t_1} = T^{t_2}$ for all $t_i \in [0, 1]$ and the height functions h^t change continuously with respect to the l_∞ norm, it follows that the path defined by $t \mapsto (T^t, h^t)$ in the space of trees is continuous. □

Definition 5.19. We say that a generic merge tree (T, h) with $n + 1$ leaves is in *standard form* if its height function h maps its leaf nodes onto $\{0, 1, \ldots, n\}$ and its internal non-root nodes onto $\{n + 1, \ldots, 2n\}$.

It is clear that a merge tree in standard form has a barcode in standard form (Definition 4.6).

Lemma 5.20. *For all $\sigma \in \mathrm{Sym}_n$, the tree realization number $R(\sigma)$ is equal to the number of combinatorial equivalence classes of merge tree whose barcode has permutation type σ.*

It follows immediately from this lemma that

$$\sum_{\sigma \in \mathrm{Sym}_n} R(\sigma) = \#\{\text{combinatorial classes of merge trees}\},$$

since barcode permutation type is also an invariant of the combinatorial equivalence type of the merge tree.

Proof. By Lemma 5.18 there is a path in $\mathcal{MT}_n$ from any merge tree whose barcode is of permutation type σ to one that is in standard form (Definition 5.19).

The tree realization number $R(\sigma)$ counts the number of merge trees in standard form with the standard form barcode $B(\sigma)$; see Definition 4.6. If two different merge trees (T, h) and (T', h') are both in standard form with the same barcode $B(\sigma)$, then they cannot be combinatorially equivalent. The inductive construction that created T and T' must have differed in a choice for some $i \in \{1, \dots, n\}$ of where to attach a branch with leaf node at height i: to a branch with leaf node at height j or height j', with $0 \leq j \neq j' < i$. An isomorphism of merge trees from (T, h) to (T', h') would have to exchange the order of of the leaf nodes at heights j and j', which is prohibited by the definition of combinatorial equivalence of merge trees (Definition 3.9). □

Since every merge tree is combinatorially equivalent to one in standard form, where leaf nodes are at heights $\{0, 1, \dots, n\}$, we can use this positioning to relate merge trees to maximal chains in the lattice of partitions of n, Definition 2.19.

Theorem 5.21. *Combinatorial equivalence classes of merge trees with $n+1$ leaf nodes are in bijective correspondence with maximal chains in the lattice of partitions $\mathcal{P}_n$. As a consequence, the sum of realization numbers is given by the following closed form formula:*

$$\sum_{\sigma \in \mathrm{Sym}_n} R(\sigma) = \sum_{\sigma \in \mathrm{Sym}_n} \prod_{i=1}^{n} l_i(\sigma) = \frac{(n+1)!n!}{2^n}.$$

Proof. Given a merge tree (T, h) in standard form with $n+1$ leaves, we explain first how to construct an associated maximal chain in the lattice of partitions, $\mathcal{P}_n$. We then show that every maximal chain is associated to some merge tree and that non-equivalent trees gives rise to distinct maximal chains.

Since (T, h) is in standard form, all of the merge events (bifurcations) happen after (are at greater height than) all the birth events. It follows that the sublevel set of $h : V(T) \to \mathbb{R}$ at any value in the interval $(n, n+1) \subset \mathbb{R}$ consists of $n+1$ components, corresponding to the finest partition $\mathcal{S}(T)_1 := \{0|1|2| \cdots |n\}$.

As we cross height $n+1$, the definition of the standard form implies that a merge event of two components, born at heights i and j, occurs. This merge event has the effect of coarsening the partition $\mathcal{S}(T)_1$, placing the two elements i and j into a single set of the partition. This defines the next, coarser partition $\mathcal{S}(T)_2$.

In general the i-th partition associated to the tree T is the partition of the leaf nodes into connected components at height $n + i$. At height $2n$ the sublevel set of the tree is connected, which corresponds to the top element in $\mathcal{P}_n$.

Each standard form merge tree thus gives rise to a chain of $2n$ elements in $\mathcal{P}_n$, which is obviously maximal. Moreover, from any maximal chain

$$\mathcal{U}_1 \preceq \cdots \preceq \mathcal{U}_\ell$$

in $\mathcal{P}_n$, one can always build a merge tree that realizes the chain as follows. Start by defining a filtration of the set of subsets of $[n]$, where a subset $V \subset \mathbf{n}$ enters the filtration at $n + i$, where i is the smallest index such that $V \subset U$ for some $U \in \mathcal{U}_i$. This defines a function from the set of subsets of $[n]$ (of which the geometric realization is the n-simplex) to $\mathbb{R}$. Taking the merge tree of this function as in Remark 3.5 associates a merge tree to a chain in $\mathcal{P}_n$.

Injectivity of the map from standard form merge trees to maximal chains is also clear. If two merge trees in standard form produce the same maximal chain, then their heights and adjacency relationships must be the same, i.e., they must be combinatorially equivalent.

The number of maximal chains in $\mathcal{P}_n$ was determined by Erdős and Moon [48] to be $(n + 1)!n!2^{-n}$. This number is easily understood in the setting of merge trees. First, one chooses two of the $n + 1$ connected components to merge at height $n + 1$. Then one chooses two of the remaining n connected components to merge at height $n + 2$. This process repeats until we run out of options at height $2n$. The number of ways of constructing standard form merge trees is thus

$$\binom{n+1}{2}\binom{n}{2}\cdots\binom{2}{2} = \frac{(n+1)n}{2} \cdot \frac{n(n-1)}{2} \cdots \frac{2 \cdot 1}{2} = \frac{(n+1)!n!}{2^n}.$$

□

Example 5.22. Figure 5.8 shows the lattice of partitions on the set $\{0, 1, 2\}$ together with the three possible merge trees corresponding to the maximal chains in the lattice.

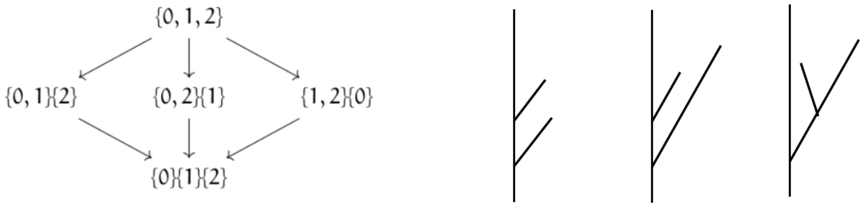


Figure 5.8: Left: The lattice of partitions of the set $\{0, 1, 2\}$. Right: The three possible merge trees corresponding to the maximal chains in the lattice, illustrating Theorem 5.21.

Remark 5.23 (Expected Tree Realization Number). It is very convenient that $n!$ appears in the numerator of the sum of realization numbers. As we explain in greater depth in the section on statistics for the realization number, this allows us to compute the expected realization number when Sym_n is equipped with the uniform measure, for which the probability of a permutation σ is $P(\sigma) = \dfrac{1}{n!}$. Indeed, by rearranging terms slightly, we see that the expected realization number is determined by the ratio of $(n+1)!$ and 2^n:

$$\mathbb{E}[R] = \sum_{\sigma \in \mathrm{Sym}_n} R(\sigma)P(\sigma) = \frac{1}{n!}\frac{(n+1)!n!}{2^n} = \frac{(n+1)!}{2^n}.$$

Before studying the probabilistic aspects of the realization number more fully, we first compare Theorem 5.21 with analogous counting results for phylogenetic trees in the next section.

5.2.5 Counting Merge Trees versus Phylogenetic Trees

In this section, we compare two counting results for combinatorial merge trees and for phylogenetic trees. On the one hand, Theorem 5.21 implies that there are $\dfrac{(n+1)!n!}{2^n}$ different combinatorial merge trees with $n+1$ leaves. On the other hand, it was shown in [49] that there are $(2n-1)!!$ distinct combinatorial phylogenetic trees with $n+1$ leaves. In general, there are more classes of merge trees than there are phylogenetic trees. In the next example, we work through the case $n = 3$ in detail.

Example 5.24. For $n = 3$, i.e., 4 leaf nodes, these formulas imply that there are 18 different classes of merge trees, but only 15 classes of phylogenetic trees, shown in Figure 5.9. In Figure 5.5C, one can see the 18 different classes of merge trees, arranged by row according to their permutation type in Sym_3. There are three pairs of merge trees highlighted with colored boxes that correspond to the same combinatorial type of phylogenetic tree.

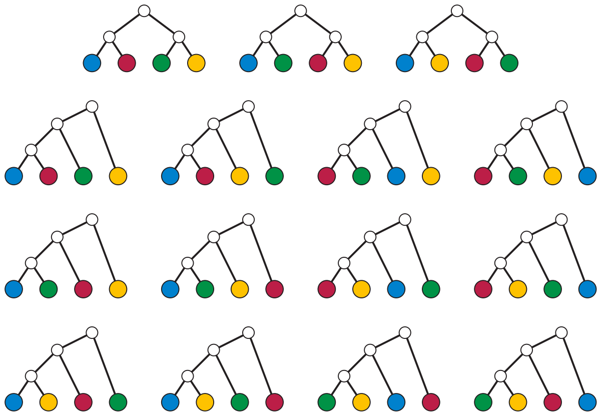


Figure 5.9: [Wikipedia, "Double Factorial", "Unordered binary trees with 4 leaves", n.d.]. The 15 different binary rooted trees with four labelled-by-color leaf nodes. The top node should be regarded as the unique child of the root node. This should be compared with the 18 different merge trees in Figure 5.5C, as discussed in Example 5.24.

As the example above shows, the essential difference between classes of merge trees and classes of phylogenetic trees is that merge trees are sensitive to relative heights of internal (death) nodes, whereas phylogenetric trees are not. This also explains why two combinatorially equivalent metric phylogenetic trees (T, m) and (T', m') may be associated to different permutation types, if one uses Proposition 3.14 to define a height function on each and compute a barcode according to the Elder rule. However there are certain orders of births and deaths that must be preserved. As one can see in Figure 5.5C, the pair of trees in the purple box under column B both have the blue bar being born before and dying after the purple bar; the relative positioning of the death time associated to the red bar is the only thing that changes.

In this section we pinpoint more precisely how many different classes of merge trees can produce the same class of phylogenetic tree. As one might imagine, this is dictated in part by certain subgroups of the symmetric group, determined essentially by the number of incomparable internal nodes in the natural partial order on the tree nodes specified by $p < q$ if p is on the unique path from q to the root. Our bound on the number of classes of merge trees that define the same class of phylogenetic trees is formulated as follows. Recall that we assume that the root of any rooted tree has a unique child.

Proposition 5.25. *Let T be a combinatorial phylogenetic tree. Let c denote the unique child of the root vertex. Let A_i be the set of internal nodes of T that are i*

hops away from c in the path metric (in particular, $A_0 = \{c\}$).

If $\eta(T)$ denotes the number of combinatorial equivalence classes of merge trees indistinguishable from T when regarded as combinatorial phylogenetic trees, then

$$\eta(T) \geq \prod_{j=0}^{k} |A_j|!.$$

Proof. We prove our result by induction on the maximum path distance in T from the child c. If the maximum path distance to the child is 0, then T has a unique internal node c, i.e., T has three nodes: the root r, its child c, and two leaves. This tree admits unique combinatorial merge and phylogenetic strucures, whence $\eta(T) = 1 = 0!$.

Suppose now the result holds whenever the maximal path distance from the child c is less than k, for some $k \geq 1$. Decompose the internal nodes of T into k sets $A_1, A_2, ..., A_k$. All nodes in A_k have only (two) leaf descendents, as otherwise there would exist an internal node further away from c than some node in A_k, so the maximal path distance to c would be greater than k.

Let $A_k = \{q_1, q_2, ..., q_s\}$. If we remove the leaf nodes attached to each $q_i \in A_k$, we obtain a phylogenetic tree T' with internal nodes partitioned into sets $A_1, A_2, ..., A_{k-1}$. By the induction hypothesis, there are at least $\prod_{j=1}^{k-1} |A_j|!$ combinatorial equivalence classes of merge trees indistinguishable from T' when considered as phylogenetic trees.

For each such equivalence class, we can obtain merge trees indistinguishable from T as phylogenetic trees by reattaching the leaves to each q_i and choosing any ordering on A_k, which we may do because all q_i are at the same distance from c, and hence are incomparable nodes. Since there are $|A_k|!$ possible total orders on the set of q_i, we can conclude. □

5.2.6 A Lattice-Theoretic Perspective on the Persistence Map

To end this section, we characterize the persistence map from combinatorial merge trees to combinatorial barcodes in terms of monotone maps between two lattices: the subset lattice and the partition lattice. We show that a maximal chain in the subset and partition lattices corresponds to a combinatorial barcode and combinatorial merge tree respectively, and that one may incrementally construct solutions to the inverse problem using this correspondence.

In this section, we denote a combinatorial barcode (i.e., the permutation associated to it) by $B = \{(i, j)\}$ if the i^{th} birth endpoint is matched with the j^{th} death endpoint.

Let $(P, \preccurlyeq)$ be a poset. Recall that a totally ordered subset $\mathcal{C} \subseteq P$ is called a *chain*. An *interval* is a subset $\mathcal{I} \subseteq P$ where if $p, q \in \mathcal{I}$ and $p \preccurlyeq r \preccurlyeq q$, then

$r \in \mathcal{I}$. A *path* γ is a chain that is also an interval. A path is *based* at $x_0 \in P$ if the lowest element in γ is x_0. If P has a unique lowest element $\hat{0}$ (e.g. a lattice), we write $\tilde{P}$ as the poset of paths based at $\hat{0}$, which is a poset via containment of paths. There is a unique surjective map $\pi_P : \tilde{P} \to P$ sending a path to its endpoint. Furthermore, if $f : P \to Q$ is a monotone map of posets, there is a unique map $\tilde{f} : \tilde{P} \to \tilde{Q}$ such that $f \circ \pi_P = \pi_Q \circ \tilde{f}$. We call $\tilde{f}$ the *lift* of f.

Recall from Section 2.2.1 that the subset lattice on $[n] = \{1, \dots, n\}$ is $\Pi_n = \mathcal{P}([n])$, the set of all subsets of $[n]$, including the empty set $\emptyset$, equipped with the partial order $\subseteq$ of "being a subset of". The poset of paths in Π_n based at $\emptyset$ is $\tilde{\Pi}_n$. A partition of the set $\mathbf{n} := \{0, 1, \dots, n\}$ is a collection of disjoint subsets $\mathcal{U} = \{U_1, \dots, U_k\}$ of $\mathbf{n}$ whose union is $\mathbf{n}$. A partition $\mathcal{U}$ *refines* a partition $\mathcal{U}'$, written $\mathcal{U} \preceq \mathcal{U}'$, if every subset of $\mathcal{U}'$ is equal to a union of elements of $\mathcal{U}$. Recall that this forms the lattice of partitions $\mathcal{P}_n$. The poset of paths based at the finest partition of $\mathbf{n}$, $\{\{0\}, \dots, \{n\}\}$, is $\tilde{\mathcal{P}}_n$.

We can filter a combinatorial barcode B with n bars into sets $B_1 \subset \cdots \subset B_n := B$ where B_k is the set of pairs $\{(i,j)\}_{j \leq k}$. We refer to B_k as a *partial (combinatorial) barcode*. The set of all partial barcodes with at most n bars forms a poset by containment, which we denote by $\mathcal{PCB}_n$. Similarly, a *partial (combinatorial) merge tree* is a filtration of a combinatorial merge tree T with $n+1$ leaves by subgraphs $T_0 \subset T_1 \subset \cdots \subset T_n := T$ where T_k is the full subgraph supported on the set of leaf nodes and all internal nodes with label less than or equal to k. Partial merge trees also forms a poset by subgraph containment, denoted $\mathcal{PCT}_n$; see Figure 5.10. The persistence map between combinatorial merge trees and barcodes extends to a map from $\mathcal{PCT}_n$ to $\mathcal{PCB}_n$, which we also call the persistence map.

Theorem 5.26. *The poset of partial merge trees $\mathcal{PCT}_n$ and barcodes $\mathcal{PCB}_n$ are isomorphic to $\tilde{\mathcal{P}}_n$ and $\tilde{\Pi}_n$, respectively. Furthermore, there is a monotone map $H : \mathcal{P}_n \to \Pi_n$ whose lift $\tilde{H} : \tilde{\mathcal{P}}_n \to \tilde{\Pi}_n$ is naturally isomorphic to the persistence map from $\mathcal{PCT}_n \to \mathcal{PCB}_n$.*

Proof. Every partial merge tree $T_0 \subset \cdots \subset T_k$ defines a path $\mathcal{U}_0 < \cdots < \mathcal{U}_k$, where $\mathcal{U}_i$ is the partition of the leaf node labels induced by connected components in the graph T_i; see Figure 5.10. Every partial barcode $B_1 \subset \cdots \subset B_k$ defines a path $\emptyset := A_0 \subset \cdots \subset A_k$, where A_k is the set of birth labels whose deaths occur by time k. These specify the isomorphisms.

Define $H : \mathcal{P}_n \to \Pi_n$ as follows: Let $(U_1, U_2, ..., U_k)$ be a partition of $\mathbf{n}$. For each U_i, let $U_i' := U_i \setminus \{\min\{x \in U_i\}\}$. Let $H((U_1, U_2, ..., U_k)) = \cup_{i \in [k]} U_i' \in \Pi_n$. This map is monotone, since if $(U_1, U_2, ..., U_k) \leq (V_1, V_2, ..., V_l)$, then the latter partition is obtained by collapsing parts of the first, which can only add elements to $H((U_1, U_2, ..., U_k))$. It is easy to see that this map is also surjective. This lifts to a natural map $\tilde{H}$, defined on paths.

The maximal element (endpoint) of a path $\gamma \in \tilde{\mathcal{P}}_n$ corresponds to a partition

$(U_1, U_2, ..., U_k)$ that indexes the leaf labels of the connected components of T_k, the k^{th} stage in a partial merge tree.

The Elder Rule (Section 3.3.1) maps each of the U_i to U_i' as $\min U_i$ encodes the oldest leaf node, which goes unpaired by the persistence algorithm. The image is the union $B = \cup_{i \in [k]} B_i$ of leaf node labels that have been killed by stage k.

The combinatorial barcode is encoded by the successive differences between B_i and B_{i+1}. □

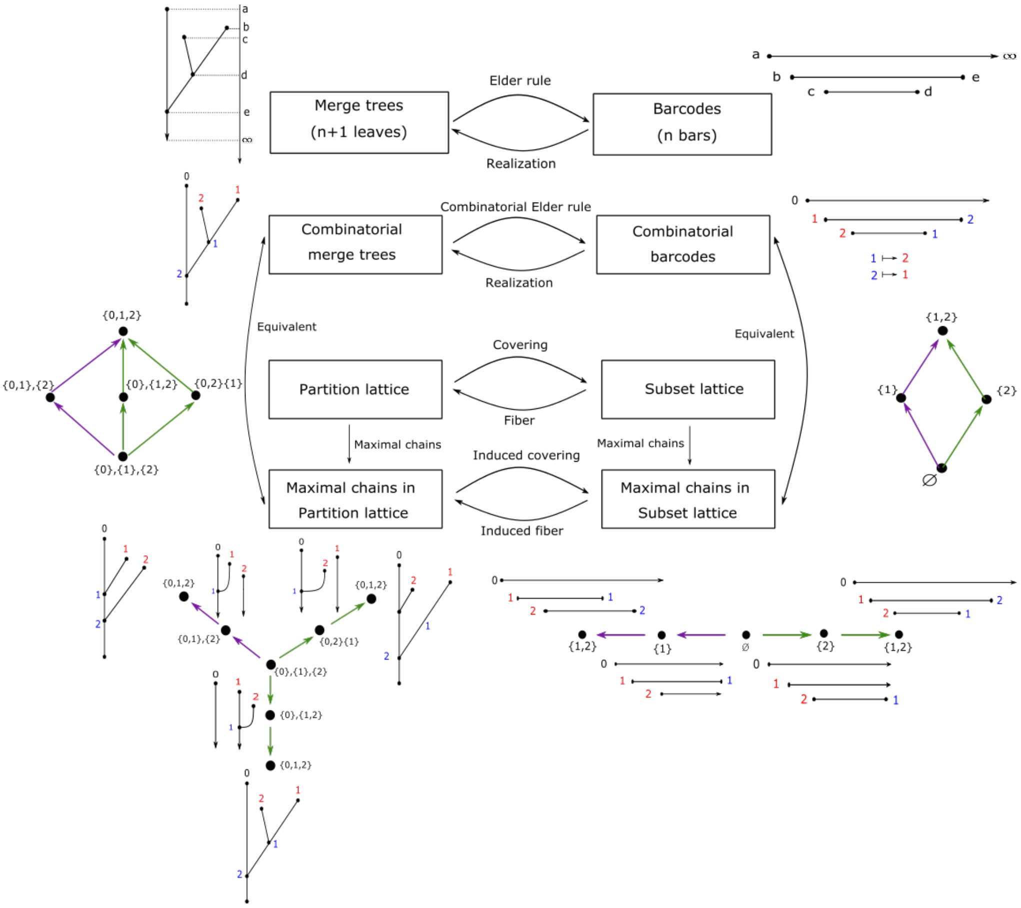


Figure 5.10: Illustration of Theorem 5.26. On the left hand side, combinatorial merge trees and the correspondence with maximal chains in the partition lattice. On the right hand side, combinatorial barcodes and the correspondence with maximal chains in the subset lattice. The inverse problem can be expressed using the lattice covering and the induced covering on the maximal chains posets.

5.3 Statistical and Probabilitstic Perspectives

5.3.1 Probabilistic Study of the TRN

As already foreshadowed by Remark 5.23, the formula in Theorem 5.21 provides us with an unexpected gift in the study of statistics for realization numbers. Assuming that every combinatorial type of barcode is equally likely, so that each permutation type σ has probability $\frac{1}{n!}$, we calculated that the expected tree realization number (TRN) is

$$\mathbb{E}[R] = \sum_{\sigma \in \mathrm{Sym}_n} R(\sigma)P(\sigma) = \frac{1}{n!}\frac{(n+1)!n!}{2^n} = \frac{(n+1)!}{2^n}.$$

We regard the assumption that each barcode permutation type is equally likely as a sort of "null hypothesis" to be tested against. Even if one considers Gaussian perturbations to data, characterizing the image of this measure on the space of merge trees and hence (combinatorial types) of barcodes is an open problem. Depending on the setup, it may be the case that features tend to die in the order in which they are born (a sort of "topological first in first out" queue) or it might be the case that features die in the opposite order in which they are born (a "first in last out" queue). In general, for real data, it is unlikely that the distribution of permutation types of (barcodes of) merge trees will be uniform. Regardless, characterizing the distribution of TRNs in terms of the output of the function $R : \mathrm{Sym}_n \to \mathbb{N}$ when Sym_n is equipped with the uniform measure provides an important null hypothese against which to test real data.

In this section we start with a brief outline of computational methods for generating random barcodes and compare the corresponding distribution of permutation types with the uniform distribution. We then provide formulas for first and second moments of the pushforward distribution $\pi_n := R_*\mu_n$, where μ_n is the uniform measure on Sym_n. This allows us to calculate the variance of the TRN, which opens the door to hypothesis testing whenever the map from trees to barcodes is of interest to scientific applications.

Somewhat surprisingly, Theorem 5.27 says that the exact value for the measure π_n can be determined from π_{n-1} and Dirichlet convolution with the uniform distribution on S_{n-1}, enabling us to study the entire distribution of TRNs as the number of features varies. To conclude, we provide a novel closed-form formula for the expected log-realization number, which allows us to characterize the empirical data in Figure 1.2 in a more analytical manner.

Let μ_n denote the uniform distribution on Sym_n. By our correspondence, this is also a distribution on combinatorial equivalence classes of barcodes. The tree realization number $R : \mathrm{Sym}_n \to \mathbb{N}$ then defines a random variable where the probability $P(R = t)$ is determined by the number of permutations n_t with realization number t. The following theorem states that this probability can be computed recursively via convolution with the uniform distribution on $1,\ldots,n$.

Theorem 5.27. *For any $k \geq 1$, let μ_k denote the uniform distribution on Sym_n and $\pi_n = R_*(\mu_n)$ its pushforward onto $\mathbb{N}$ via $R : \mathrm{Sym}_n \to \mathbb{N}$. Let U_k denote the uniform distribution on $\{1,2,...,k\}$.*

The probability mass function of π_n can be recursively computed as follows.

- *$\pi_1 = U_1$.*
- *For $k > 1$, $\pi_k = U_k * \pi_{k-1}$, where $*$ indicates Dirichlet convolution, i.e,.*

$$\pi_k(c) = \sum_{ab=c} U_k(a)\pi_{k-1}(b) \text{ for all } c \in \mathbb{N}.$$

It follows immediately from this theorem that

$$\pi_n = U_n * U_{n-1} * ... * U_1$$

for all $n \geq 1$.

Proof. We prove this theorem by induction on k. It holds trivially for $k = 1$. Suppose that it holds for $k-1$ for some $k \geq 2$. Each number that has positive probability under π_{k-1} corresponds to $R(\sigma) = \prod_{i=1}^{k-1} l_i(\sigma)$ for some $\sigma \in \mathrm{Sym}_{k-1}$.

Consider the map $\kappa_{k-1}^j : \mathrm{Sym}_{k-1} \to \mathrm{Sym}_k$ that embeds Sym_{k-1} into Sym_k as follows. For every $\sigma \in \mathrm{Sym}_{k-1}$, the permutation $\kappa_{k-1}^j(\sigma)$ is specified by

$$\kappa_{k-1}^j(\sigma)(i) = \begin{cases} j & \text{if } i = k \\ \sigma(i) + 1 & \text{if } \sigma(i) \geq j \\ \sigma(i) & \text{if } \sigma(i) < j. \end{cases}$$

In other words, κ_{k-1}^j sends $\sigma \in \mathrm{Sym}_{k-1}$ to the permutation $\kappa_{k-1}^j(\sigma) \in \mathrm{Sym}_k$ that maps the k-th object to j and then "bumps up" by one the assigned value of elements in $\{1,2,...,k-1\}$ that are mapped to an element greater than or equal to j.

Each map in the collection $\{\kappa_{k-1}^j\}_{j=1}^k$ is injective and collectively their images surject onto Sym_k. To determine the realization numbers for Sym_k, we therefore need only compute the realization number of $\kappa_{k-1}^j(\sigma)$ for all $j \in \{1,\ldots,k\}$ and $\sigma \in \mathrm{Sym}_{k-1}$.

Consider $R\big(\kappa_{k-1}^j(\sigma)\big) = \prod_{i=1}^k l_i\big(\kappa_{k-1}^j(\sigma)\big)$. Since

$$l_i(\sigma) = |\{r \leq i \mid \sigma(r) \geq \sigma(i)\}|$$

for any permutation σ, it follows that

$$l_i\big(\kappa^j_{k-1}(\sigma)\big) = l_i(\sigma)$$

for all $i < k$. On the other hand, since $r \leq k$ for all $r \in \{1, .., k\}$,

$$|\{r \leq k \mid \kappa^j_{k-1}(\sigma(r)) \geq \kappa^j_{k-1}(\sigma(k))\}| = |\{r \mid \sigma(r) \geq j\}| = k - j + 1.$$

We conclude that $R\big(\kappa^j_{k-1}(\sigma)\big) = (k - j + 1)R(\sigma)$.

By the construction of κ^j_{k-1},

$$\mu_k = \frac{1}{k}\sum_{j=1}^{k}(\kappa^j_{k-1})_*(\mu_{k-1}),$$

where $(\kappa^j_{k-1})_*(\mu_{k-1})$ is the pushforward of μ_{k-1} by κ^j_{k-1}, since each pushforward assigns mass $\frac{1}{(k-1)!}$ to each element of a unique subset of size $(k-1)!$ in Sym_k.

We are now prepared to compute π_k. Let $x \in \mathbb{N}$.

$$\begin{aligned}
\pi_k(x) &= R_*(\mu_k)(x) = \mu_k\big(R^{-1}(x)\big) && (5.1)\\
&= \frac{1}{k}\sum_{j=1}^{k}(\kappa^j_{k-1})_*(\mu_{k-1})\big(R^{-1}(x)\big) && (5.2)\\
&= \frac{1}{k}\sum_{j=1}^{k}\mu_{k-1}\Big((\kappa^j_{k-1})^{-1}\big(R^{-1}(x)\big)\Big) && (5.3)\\
&= \frac{1}{k}\sum_{j=1}^{k}\mu_{k-1}\Big((\kappa^j_{k-1})^{-1}\big(\{\sigma \in \mathrm{Sym}_k \mid R(\sigma) = x\}\big)\Big) && (5.4)\\
&= \frac{1}{k}\sum_{j=1}^{k}\mu_{k-1}\big(\{\tilde{\sigma} \in \mathrm{Sym}_{k-1} \mid (k-j+1)\cdot R(\tilde{\sigma}) = x\}\big) && (5.5)\\
&= \frac{1}{k}\sum_{j=1}^{k}\mu_{k-1}\big(\{\tilde{\sigma} \in \mathrm{Sym}_{k-1} \mid j \cdot R(\tilde{\sigma}) = x\}\big) && (5.6)\\
&= \frac{1}{k}\sum_{jb=x}\mu_{k-1}\big(\{\tilde{\sigma} \in \mathrm{Sym}_{k-1} \mid R(\tilde{\sigma}) = b\}\big)\mathbb{1}_{[k]}(j) && (5.7)\\
&= \sum_{ab=x}\mu_{k-1}\big(\{\tilde{\sigma} \in \mathrm{Sym}_{k-1} \mid R(\tilde{\sigma}) = b\}\big)\frac{\mathbb{1}_{[k]}(a)}{k} && (5.8)\\
&= \sum_{ab=x}U_k(a)\pi_{k-1}(b), && (5.9)
\end{aligned}$$

where the second line follows from the identity $\mu_k = \frac{1}{k}\sum_{j=1}^{k}(\kappa_{k-1}^{j})_*(\mu_{k-1})$, the fifth line follows from $R(\kappa_{k-1}^{j}(\sigma)) = (k-j+1)R(\sigma)$, and the sixth and seven lines are simple changes of variables. □

For what follows, it is useful to consider for each n the multiset Π_n, which is the range of $R : \mathrm{Sym}_n \to \mathbb{N}$, taking into account multiplicities. Let $m_n : \mathbb{N} \to \mathbb{Z}_{\geq 0}$ be the multiplicity function of Π_n, i.e., $m_n(x)$ is the number of times $x \in \mathbb{N}$ appears in Π_n, which is the number of permutations in Sym_n that have realization number x. In particular, $m_n(x) = 0$ if and only if $x \notin \Pi_n$.

Since π_n is the pushforward of the uniform distribution on Sym_n, the probability of each x is determined by dividing the multiplicity function by $n!$, i.e., $\pi_n(x) = \frac{m_n(x)}{n!}$. The following corollary follows directly from the construction of π_n.

Corollary 5.28. *The multiset Π_n can be constructed recursively as follows:*

- $\Pi_1 = \{1\}$.
- *For $i > 1$, Π_i is the multiset with multiplicity function specified by*

$$m_n(x) = \sum_{ab=x} m(b)\mathbb{1}_{[i]}(a)\mathbb{1}_{\Pi_{i-1}}(b).$$

In other words, Π_n can be defined as a $[k] * \Pi_{n-1}$, where $[k] = \{1, \ldots, k\}$ and $*$ is the Dirichlet convolution of multisets.

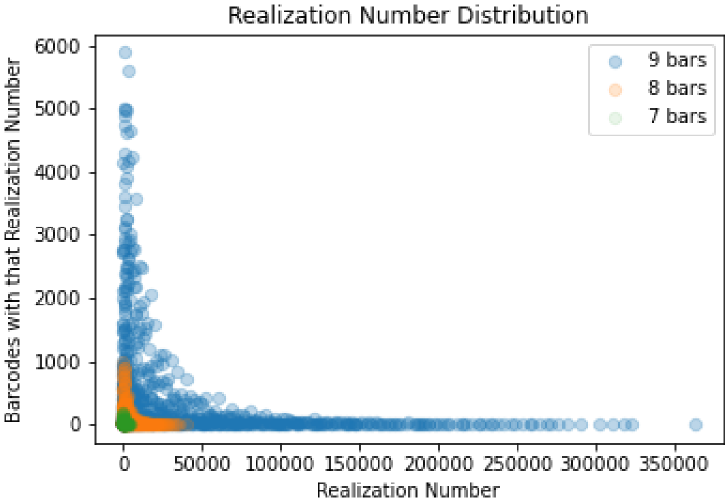


Figure 5.11: Distribution of Realization Numbers for 7,8,9 bars.

Example 5.29. We now explicitly describe Π_i for $i \in \{1,2,3,4\}$. For convenience we write the mutisets Π_i as sets with repetition. Counting the number of appearances of a number k determines $m_i(k)$.

$$\begin{aligned}
&\Pi_1 = \{1\}\\
&\Pi_2 = \{1,2\}\\
&\Pi_3 = \{1,2,2,4,3,6\}\\
&\Pi_4 = \{1,2,2,2,3,3,4,4,4,4,6,6,6,6,8,8,8,9,12,12,12,16,18,24\}
\end{aligned}$$

To conclude this section, we consider the moments of π_n. We explicitly calculate its first and second moments, obtaining the mean and variance of π_n as corollaries, and outline a general formula for the higher moments. Recall that by knowing these two moments one can use Chebyshev's inequality to calculate the probability of observing some deviation from the mean, an important first pass at hypothesis testing.

Proposition 5.30. $\mathbb{E}(\pi_n) = \frac{(n+1)!}{2^n}$.

Proof. This is the content of Remark 5.23, which establishes this proposition as a consequence of Theorem 5.21. □

Proposition 5.31. $\mathbb{E}(\pi_n^2) = \frac{(n+1)(2n+1)!}{12^n}$.

Proof. We prove the result by induction on n. The base case ($n = 1$) holds trivially, so assume that the formula holds for $n = k$.

Consider $\mathbb{E}(\pi_{k+1}^2) = \sum_{b\in B_n} R(b)^2$. Since

$$(k+1)!\sum_{b\in B_n} R(b)^2 = (k+1)!\mathbb{E}(\pi_{k+1}^2) = \sum_{x\in\mathbb{N}} m_{k+1}(x)^2,$$

to prove our result, we need only show that

$$\sum_{x\in\mathbb{N}} m_{k+1}(x)^2 = \frac{(n+1)!(2n+1)!}{12^n}.$$

We call the quantity on the left $\mathbb{E}(\Pi_{k+1}^2)$:

$$\mathbb{E}(\Pi_{k+1}^2) = \sum_{b\in B_{k+1}} R(b)^2 = \sum_{a=1}^{k+1}\sum_{b\in B_k}(aR(b))^2 = \sum_{a=1}^{k+1} a^2 \sum_{b\in B_k} R(b)^2 = \sum_{a=1}^{k+1} a^2\mathbb{E}(\Pi_k^2).$$

By the sum of squares formula, we can rewrite this as

$$\left(\frac{(k+1)(k+2)(2k+3)}{6}\right)\mathbb{E}(\Pi_k^2)$$

$$= \left(\frac{(k+1)(k+2)(2k+3)(2k+2)}{(2k+2)6}\right)\mathbb{E}(\Pi_k^2)$$
$$= \left(\frac{(k+2)(2k+2)(2k+3)}{2*6}\right)\mathbb{E}(\Pi_k^2)$$
$$= \left(\frac{(k+2)(2k+2)(2k+3)}{12}\right)\left(\frac{(k+1)!(2k+1)!}{12^k}\right)$$
$$\frac{(k+2)!(2k+3)!}{12^{k+1}} = \frac{((k+1)+1)!(2(k+1)+1)!}{12^{k+1}}.$$

□

Corollary 5.32. *The variance of π_n is*

$$\mathbb{V}(\pi_n) = \mathbb{E}(\pi_n^2) - \mathbb{E}(\pi_n)^2 = \frac{1}{n!}\left(\frac{(n+1)!(2n+1)!}{12^n} - \frac{(n!(n+1)!)^2}{n!4^n}\right).$$

Remark 5.33 (Higher Moments of the TRN). In general, we can define the k-th moment $\mathbb{E}(\pi_n^k)$ by rewriting $n!\mathbb{E}(\pi_n^k) = \mathbb{E}(\Pi_n^k) = (\sum_{a=1}^n a^k)\mathbb{E}(\Pi_n^{k-1})$ and using this recursive relationship to compute a formula. We note that by Faulhaber's formula,

$$\sum_{a=1}^n a^k = \sum_{i=0}^k \frac{(-1)^{k-i}}{i+1}\binom{k}{i}B_{k-i}n^{i+1},$$

where B_{k-i} is the $k-i$ Bernoulli number.

One can view the results above as a complete characterization of TRNs under the null hypothesis that combinatorial classes of barcodes are distributed uniformly or as part of the growing literature on statistics on the symmetric group, see e.g., [71]. We now investigate another such statistic.

Since the maximum realization number for a barcode with n non-essential bars is $n!$, it is convenient to work instead with the logarithm of the realization number, which we call the log realization number. The log realization number is used in Chapter [67] as a statistic on barcodes obtained from dendrites; see Figure 1.2 for a reminder. In particular, we can distinguish between apical and basal dendrites. Of course, the process of taking the logarithm affects the distribution of TRNs. Jensen's inequality provides a way to bound the expected log realization number. In this section we compute the expected log realization number of uniformly drawn barcodes.

Proposition 5.34. *The expected log realization number for a combinatorial class B of barcodes drawn from the uniform distribution on* Sym_n *is*

$$\mathbb{E}_{\mu_n}\Big(\log\big(R(B)\big)\Big) = \sum_{i=1}^n \frac{\log(i!)}{i}.$$

Proof. Recall that the set of left inversion vectors can be coordinatized as $[1] \times [2] \times ... \times [n]$. Since this Cartesian product has size $n!$, a uniform distribution on Sym_n can be viewed as a uniform distribution on the set of left inversion vectors. Let the notation $\mathbb{P}(B \sim \mu_n)$ denote the probability of a combinatorial equivalence class of barcodes B under the uniform distribution, that is $\frac{1}{n!}$. It follows that

$$\begin{aligned}
\mathbb{E}_{\mu_n}\Big(\log\big(R(B)\big)\Big) &= \sum_{B\in \mathrm{Sym}_n} \log(\prod_{i=1}^{n} l_i(B))\mathbb{P}(B \sim \mu_n) \\
&= \frac{1}{n!}\sum_{B\in[1]\times...\times[n]}\sum_{i=1}^{n}\log(l_i(B)) \\
&= \frac{1}{n!}\sum_{i=1}^{n}\sum_{B\in[1]\times...\times[n]}\log(l_i(B)).
\end{aligned}$$

Since $B \sim \mu_n$, and each coordinate in $[1] \times [2] \times ... \times [n]$ is independent, the interior sum (for fixed i) is equal to $\frac{n!}{i}(\log(1) + \log(2) + ... + \log(i))$. Hence

$$\mathbb{E}_{\mu_n}(\log(R(b))) = \sum_{i=1}^{n}\frac{\log(1) + \log(2) + ... + \log(i)}{i} = \sum_{i=1}^{n}\frac{\log(i!)}{i}.$$

□

Chapter 6
A Biological Inverse Problem

This chapter concerns a different inverse problem relating trees and barcodes. While the previous chapter considered the "real" inverse problem of how many combinatorial trees have the same barcodes and how to build these trees from a given barcode, here we investigate a stochastic inverse, the *topological neuronal synthesis* (TNS) algorithm [67]. The TNS takes as input a barcode B and returns a single tree T. However, the barcode of TNS(B) needs not be the same as B, due to the stochasticity of the TNS. The TNS was used in [67] to build artificial populations of neurons based on biological trees.

In this chapter, we study the composite of the TNS and TMD algorithms from a theoretical perspective, to quantify the extent to which the TNS acts as an inverse to the TMD. For a given barcode B, we show that, for a reasonable choice of parameter in the TNS, the probability that the bottleneck distance between the barcodes B and $\mathrm{TMD} \circ \mathrm{TNS}(B)$ is greater than ε decreases with ε, thus establishing a form of stability for the TNS. We prove, moreover, that the probability that two bars of a barcode B will be permuted by applying $\mathrm{TMD} \circ \mathrm{TNS}$ decreases exponentially with the distance between the terminations of the two bars, which is another form of stability. Together these stability results imply that the TNS is an excellent approximation to a (right) inverse to the TMD.

Finally, we present computational results that illustrate the complex relationship between a barcode and its possible tree-realizations. In particular, we study the distinguishing characteristics of "biological" geometric trees, i.e., those that arise from digital reconstructions of neurons, as opposed to arbitrary geometric trees. We also show that both the combinatorial type and the TMD-type of a geometric tree can change significantly when applying the composite $\mathrm{TNS} \circ \mathrm{TMD}$, from which it follows that the TNS is not a left inverse to the TMD.

This chapter is based on joint work with K. Hess and L. Kanari [69].

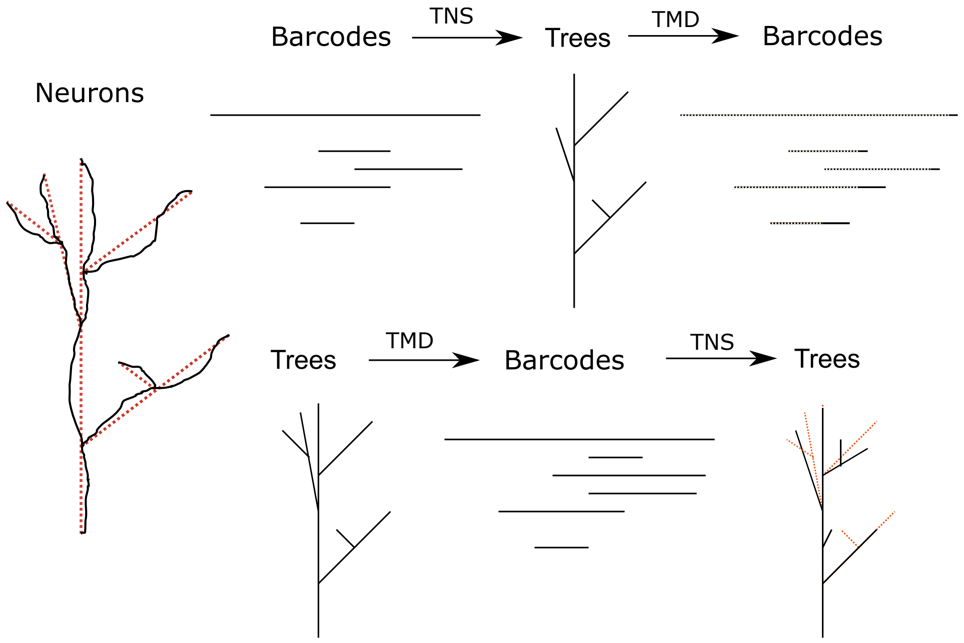


Figure 6.1: The two composites of TMD and TNS. (Left) An illustration of how a neuron (black) is modeled as a tree (dashed red lines). Recall that we describe this process and how to extract a barcode from a tree in section 3.9. (Top) The composite TMD $\circ$ TNS applied to a barcode B. The new barcode $B' = \text{TMD} \circ \text{TNS}(B)$ is indicated in dashed lines on top of the barcode B on the right. We show in section 6.2 that the barcodes B and B' will almost certainly be very similar and quantify this similarity. (Bottom) The composite TNS $\circ$ TMD applied to a tree T. The tree T that we start with is indicated in dashed red lines under the new tree $T' = \text{TNS} \circ \text{TMD}(T)$. The trees T and T' can be quite different combinatorially, as seen on the right.

Remark 6.1 (Different Notation for Births and Deaths). This chapter follows the convention of [67–69] and considers persistence diagrams in a slight different way than in the rest of this thesis. Bifurcations in trees usually correspond to the death times in the barcodes and the termination to the birth times (see Figure 3.2 for instance). However, in [67–69], motivated by applications in the study of neuron populations, births and deaths' roles are switched so that the births correspond to bifurcations and deaths to terminations of branches. Hence, persistence diagrams are "upside down" with respect to the usual convention in TDA: the birth times/terminations are on the y-axis and the death times/bifurcations are on the x-axis, resulting in points of the persistence diagrams being under the diagonal. Note that switching the role of birth and death times does not influence the tree realization number of the barcode. The notation for this chapter are summarized in the box below.

Strict barcode: a barcode $B = \{(b_i, d_i)\}_{i\in\{0,\ldots,n\}}$ such that the essential bar (b_0, d_0) contains all the others bars and $b_i \neq b_i, d_i \neq d_j$ if $i \neq j$.
Births: they correspond to the bifurcation points of the branches, i.e., the internal nodes.
Deaths: they correspond to the termination points of the branches, i.e., the tips of the branches.
Permutation: it is computed from a barcode B with ordered birth times via $\sigma_B(i) = \#\{j < i \mid d_j \leq d_i\}$. The essential bar (b_0, d_0) is not considered.
The space $\mathcal{B}_n$: it consists of all barcodes with $n + 1$ bars, including the essential one (b_0, d_0).

6.1 The TNS: A Stochastic Inverse

The topological neuron synthesis (TNS) algorithm [67] stochastically generates synthetic neurons, in particular for use in digital reconstructions of brain circuitry [80]. In this thesis, we focus on the sub-process of the TNS that stochastically generates a geometric tree from a strict barcode, in such a way that if a tree T is generated from a barcode B, then TMD(T) is "close to" to B, with respect to the bottleneck metric on the set of barcodes, up to some stochastic noise, c.f. Section 6.2. Henceforth, when we refer to the TNS, we mean this sub-process.

We summarize the TNS algorithm below, following [67]. To grow geometric trees, the TNS algorithm first initiates growth, then loops through steps of *elongation* and *branching/termination*. Each branch of the tree is elongated as a directed random walk [7] with memory. At each step, a growing tip is assigned probabilities to bifurcate, to terminate, or to continue that depend on the path distance from the root and on a chosen bar of the selected barcode. Once a bar has been used, it is removed from the barcode. The growth of a tree terminates when no bars remain to be used. We now provide further details of the two steps in this process.

Bifurcation / Termination

The branching process in the TNS algorithm is based on the concept of a *Galton-Watson tree* [55], which is a finite rooted tree recursively generated as follows. At each step, a number of offspring is independently sampled from a distribution. Since a geometric tree consists only of bifurcations, terminations, and continuations, the accepted values for the number of offspring are: zero (termination), one (continuation), and two (bifurcation). The Galton-Watson algorithm generates only a combinatorial tree, with no embedding in space, so the traditional process is modified to introduce a dependency of the tree growth on the embedding, so that the bifurcation/termination probabilities depend on the path distance of the growing tip from the root.

The bifurcation/termination step of the growth process of a geometric tree with associated barcode B proceeds as follows. Each growing tip of the tree is assigned a bar (b_i, d_i) sampled from the barcode B and a bifurcation angle a_i. The growing tip first checks the probability to bifurcate, then the probability to terminate. If the growing tip does not bifurcate or terminate, then the branch continues to elongate. The probability to bifurcate depends on b_i: as the distance from the root to the growing tip approaches b_i, the probability to bifurcate increases exponentially until it attains a maximum of 1 at b_i. Similarly, the probability to terminate depends exponentially on d_i.

The probabilities to bifurcate and terminate are sampled from an exponential distribution $e^{-\lambda x}$, whose free parameter λ should be wisely chosen. A very steep exponential distribution (high value of λ) reduces the variance of the population of geometric trees synthesized based on the same barcode. On the other hand, a very low value of λ results in trees that are almost random, since the dependence on the input persistence barcode is decreased significantly. If we assume that growth takes place in discrete steps of size L, the value of the parameter λ should be of the order of the step size L, to ensure biologically appropriate variance [67]. Assuming $L = 1$ in some appropriate units, we usually select $\lambda \approx 1$, so that the bifurcation and termination points are stochastically chosen but still strongly correlated with the input persistence barcodes.

The Elder Rule and TNS

The TNS provides a sort of right inverse to the TMD. To recreate a tree that is close to the original, the branch corresponding to a particular bar (b_i, d_i) in the barcode can be attached only to branches corresponding to bars (b_j, d_j) such that $d_i < d_j$ and $b_i > b_j$. This rule ensures that the Elder rule (at a bifurcation, the longer component survives) holds in the TMD transformation. As a result, only a subset of trees with n branches can be generated by the TNS from a given strict barcode with n bars.

	TMD	**TNS**
Goal	Compute the barcode of a tree based on a distance function	Grow a new tree from a barcode
Directionality	From leaves to root	From root to leaves
Domains	{geometric trees} $\longrightarrow$ {barcodes}	{barcodes} $\longrightarrow$ {geometric trees}

Table 6.1: Summary and terminology of the TMD and TNS algorithms. The TMD computes the barcode of a tree from the tips of branches towards the root, whereas the TNS grows the tree in the opposite direction, from the root to the leaves.

6.2 Stability of the TNS

In this section, we investigate the effect of the composition of the TNS and TMD algorithms from a theoretical perspective. Given a strict barcode $B = \{(b_i, d_i)\}_{i\in\{0,\ldots,n\}}$, we apply the TNS to B, for a fixed choice of the parameter λ, obtaining a tree T_B, and then compute the barcode of T_B, $\mathrm{TMD}(T_B)$, which we denote by $B' = \{(b'_i, d'_i)\}_{i\in\{0,\ldots,n\}}$. To quantify to what extent the TNS acts as an inverse to the TMD, we are interested in determining how similar B and B' are.

Expressing the similarity between B and B' in terms of the (modified) bottleneck distance (Theorem 4.23) enables us to establish one form of stability for the TNS in the first part of this section. We establish another type of stability for the TNS in the second part, when we show that the probability that the order of two specific bars will be altered upon applying TMD $\circ$ TNS decreases exponentially with the distance between the death times of the two bars.

6.2.1 Modified Bottleneck stability

We call the endpoints of the bars of the barcode B *targets*, as the TNS algorithm either creates a new branch or terminates a branch when the distance from the root approaches a birth or death point, respectively.

By definition of the TNS algorithm, when approaching a target, there is an exponential probability to bifurcate (create a new branch) or terminate, depending on λ. It follows that for any bar (b_i, d_i) of a given barcode B, the distance between b_i and b'_i (the bifurcation distance and the target bifurcation distance of the i^{th} branch) and the distance between d_i and d'_i (the termination distance and the target termination distance of the i^{th} branch) should follow an exponential distribution of parameter λ,

$$|b_i - b'_i| \sim \mathrm{Exp}(\lambda) \text{ and } |d_i - d'_i| \sim \mathrm{Exp}(\lambda).$$

The notion of similarity between barcodes that we consider here is the modified bottleneck distance, Definition 4.23, which we denote simply by d in the rest of this chapter. We recall the definition here:

Definition 6.2. Let $B = \{(b_i, d_i)\}_{i\in\{0,\ldots,n\}}$ and $B' = \{(b'_i, d'_i)\}_{i\in\{0,\ldots,n\}}$ be two barcodes in $\mathcal{B}_n$. The *modified bottleneck distance* between B and B' is

$$\tilde{d}(B, B') := \min_{\gamma\in\mathrm{Sym}_{n+1}} \max_{i\in\{0,\ldots,n\}} \|(b_i, d_i) - (b'_{\gamma(i)}, d'_{\gamma(i)})\|_\infty.$$

where $\|\cdot\|_\infty$ is the l^∞-norm on $\mathbb{R}^2$.

Lemma 6.3. *Let B be a strict barcode with n bars, and let $B' = \mathrm{TMD}\circ\mathrm{TNS}(B)$. If $B \sim B'$, then*

$$\mathbb{P}\big(d(B, B') > \varepsilon\big) \le 1 - (1 - \exp(-\lambda\varepsilon)(\lambda\varepsilon + 1))^n. \tag{6.1}$$

Proof. Considering the case where γ is the identity, we see that

$$d(B, B') \leq \sup_i |b_i - b'_i| + |d_i - d'_i|.$$

If $B \sim B'$, the differences between the new and original values of the births and deaths all follow an exponential distribution,

$$|b_i - b'_i| \sim \text{Exp}(\lambda) \text{ and } |d_i - d'_i| \sim \text{Exp}(\lambda).$$

The cumulative probability distribution function of $|b_i - b'_i| + |d_i - d'_i|$ is thus given by an Erlang$(2, \lambda)$ distribution [10]

$$\mathbb{P}\big(|b_i - b'_i| + |d_i - d'_i| \leq \varepsilon\big) = 1 - (1 + \lambda\varepsilon)\exp(-\lambda\varepsilon).$$

Because we consider the supremum over i of the sum $|b_i - b'_i| + |d_i - d'_i|$, and all of the $|b_i - b'_i| + |d_i - d'_i|$ are *i.i.d*, it follows from the theory of order statistics [4] that

$$\mathbb{P}\big(d(B, B') \leq \varepsilon\big) \geq \mathbb{P}(\sup_i |b_i - b'_i| + |d_i - d'_i| \leq \varepsilon) = (1 - \exp(-\lambda\varepsilon)(\lambda\varepsilon + 1))^n. \quad (6.2)$$

Considering the probability of the complement leads to the result in Equation 6.1. □

Lemma 6.3 implies that the TNS is stable with respect to the modified bottleneck distance, in a manner dependent on the parameter λ. To illustrate this dependence, we plot the function of Equation 6.1 for different values of λ in Figure 6.2. The curve obtained for $\lambda = 1$ (blue in Figure 6.2) makes it clear that setting $\lambda = 1$ ensures that the TNS gives rise to a diverse family of new trees that are nonetheless topologically not significantly far from the original ones, which is the desired goal from a biological perspective. Making an appropriate choice of the parameter λ is thus essential.

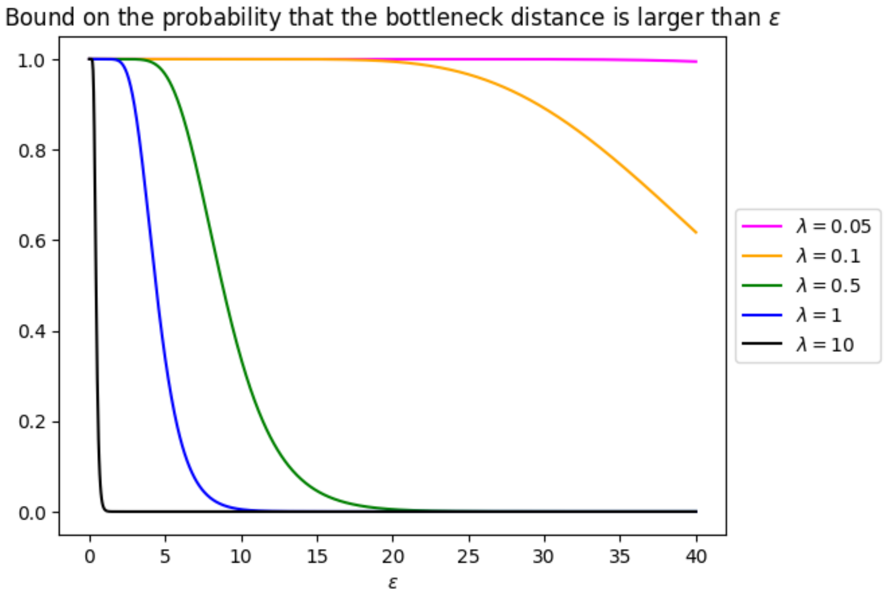


Figure 6.2: Upper bound on the probability that the modified bottleneck distance between B and $\mathrm{TNS} \circ \mathrm{TMD}(B)$ is larger than ε (Equation 6.1) for various values of λ and for $n = 10$.

If $B \sim B'$, the bound by $\gamma = \mathrm{id}$ in the formula for the modified bottleneck distance is computed between pairs of points that follow the same exponential law, as the order of bars is preserved. If $B \not\sim B'$, for example when a switch of bars occurs, then we cannot assume that the distances between matched pairs of points in the computed modified bottleneck distance follow the same law. Change of permutation type between B and B' is more frequent for small λ (Figure 6.6). Therefore, the previous lemma is usually not applicable for small values of λ, for which it is any case not particularly useful, as shown in Figure 6.2. In Figure 6.3 we summarize graphically the discussion above. The transposition of bars is studied in detail in the next section.

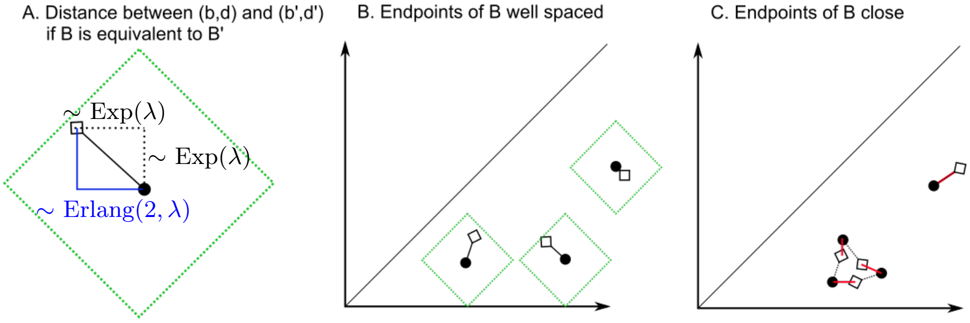

A. Distance between (b,d) and (b',d') if B is equivalent to B'
~ Exp(λ)
~ Exp(λ)
~ Erlang(2, λ)
B. Endpoints of B well spaced
C. Endpoints of B close

Figure 6.3: A. The ℓ_1-distance between the black bullet and the diamond follows an Erlang$(2, \lambda)$ distribution. The interior of the green square defines a bound for the ℓ_1-distance from the black bullet that depends on the value of the parameter λ. B. If the endpoints of the bars of B are sufficiently far away from each other and $B \sim \text{TMD} \circ \text{TNS}(B)$, then, with high probability, taking $\gamma = \text{id}$ will minimize the ℓ_1-distance between pairs of endpoints of bars. C. If the endpoints of B are instead close to each other, then it is more likely that $B \not\sim \text{TMD} \circ \text{TNS}(B)$, so that the optimal choice of γ (represented by red segments) is not the identity. The red distances do not necessarily follow exponential distributions, so the proof of Lemma 6.3 does not apply.

We perform two experiments to illustrate our theoretical results computationally. First, we compute the modified bottleneck distance between input barcodes B and output barcodes B', for increasing values of lambda λ from 0.01 to 2 (see Figure 6.4A). The computational results (average modified bottleneck distances in red, Figure 6.4A2) fit the curve of the expected mean of the probability density function[1] well (blue curve).

We also compute the cumulative density function for $0 \leq \epsilon \leq 200$ and $0 \leq \lambda \leq 2$, which we compare to the computational results (red points, Figure 6.4A3), showing that they match the theoretical prediction (blue colormap) very closely for a wide range of sufficiently large λ (zoom-in, Figure 6.4A3). However, for very small values of λ, the condition $B \sim B'$ is not always satisfied, leading to the observation that for $\lambda < 0.2$, the computationally computed modified bottleneck distances are larger than the theoretically expected values.

Second, we compute the modified bottleneck distance between input and output barcodes for various fixed values of λ, where the input barcodes arise by gradually decreasing the death time of one bar of an initial barcode B and thus increasing the distance to the next death time in the sequence (see Figure 6.4B). All other bars of the initial barcode B remain the same. We observe that the modified bottleneck distance depends only on the value of λ and not on the distance between the bars of the input barcode.

[1] The PDF can be deduced from Equation 6.2 in the proof of Lemma 6.3 by taking the derivative of $(1 - \exp(-\lambda\varepsilon)(\lambda\varepsilon + 1))^n$.

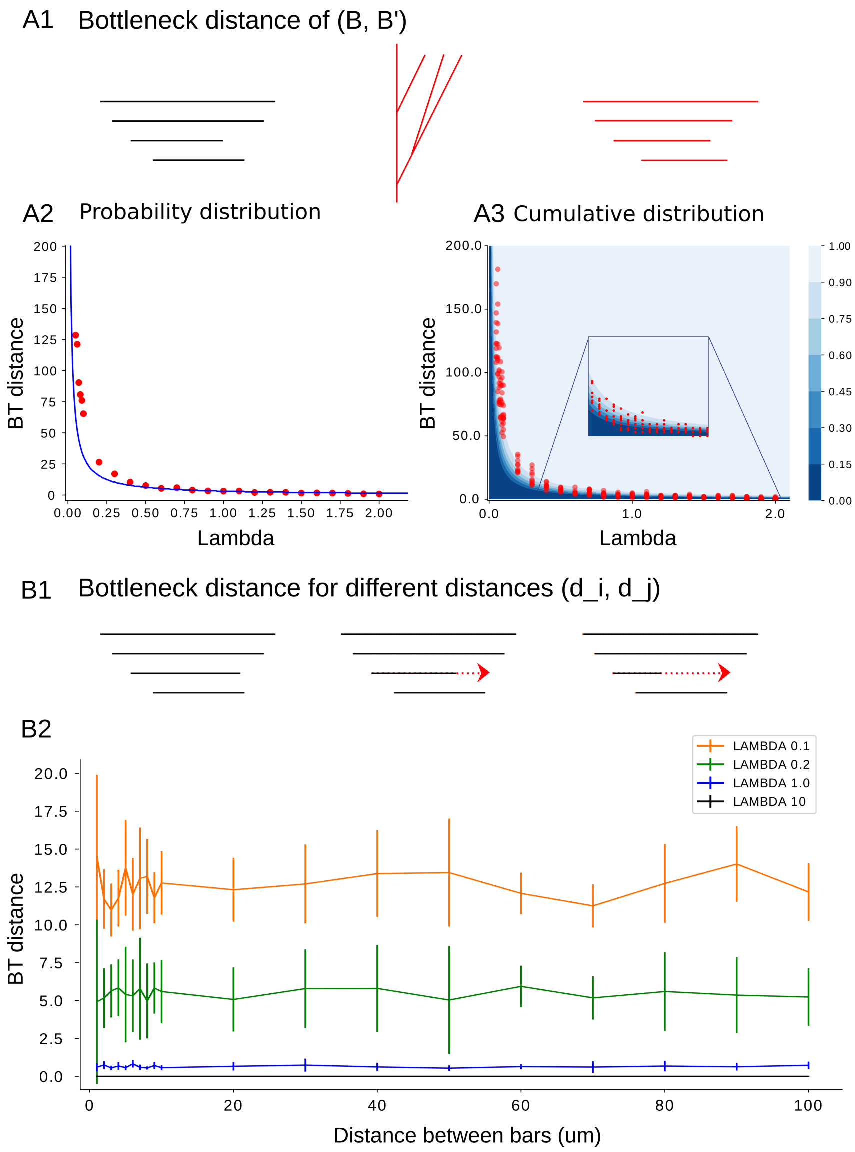


Figure 6.4: A. Modified bottleneck distance as a function of λ. We compute the modified bottleneck distance between an input barcode B and an output barcode B' for $\lambda = 0.01 - 2$. A1. From barcode B (in black), a tree (in red) is generated using the TNS which results in a new barcode $B' = \text{TMD} \circ \text{TNS}(B)$ (in red). A2. The average modified bottleneck distance (red points) is compared to the expected mean of the probability distribution function found in Lemma 6.3 (blue curve). A3. The modified bottleneck distances (red) are compared to the cumulative distribution probability for $0 < \epsilon < 200$ and $0 < \lambda < 2$ (blue). B. Modified bottleneck distance between B and B' as a function of distances between bars in B. B1. We consider barcodes of the same permutation type for different distances between two bars (b_i, d_i) and (b_j, d_j) of the initial barcode B that are consecutive in the order of deaths. B2. For each input barcode with increasing d_i, distance between death times presented in x-axis, 100 synthesized barcodes are generated and the modified bottleneck distance between the input and output barcodes is computed (y-axis), which depends only on the value of λ and not on the distance between the bars.

6.2.2 Transposition stability

As the TNS algorithm is a stochastic process, the image of any strict barcode $B = \{(b_i, d_i)\}$ under the composite TMD $\circ$ TNS essentially always differs at least slightly from B. Here we determine the probability that the orders of the death times of two specific bars of B and $\mathrm{TMD} \circ \mathrm{TNS}(B) = B' = \{(b'_i, d'_i)\}$ are different, so that B and $\mathrm{TMD} \circ \mathrm{TNS}(B)$ are not combinatorially equivalent, i.e., the associated permutations are different, as long as the birth times are not also transposed.

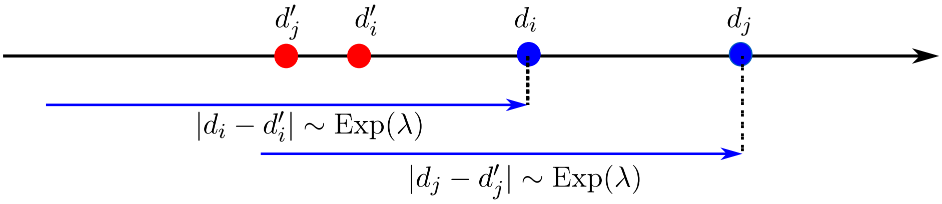

d'_j
d'_i
d_i
d_j
$|d_i - d'_i| \sim \mathrm{Exp}(\lambda)$
$|d_j - d'_j| \sim \mathrm{Exp}(\lambda)$

Figure 6.5: We are interested in the case where $d'_j < d'_i$ when we start from $d_i < d_j$. The distances $|d_i - d'_i|$ and $|d_j - d'_j|$ both follow an exponential law of parameter λ. The probability to terminate increases exponentially when approaching d_i and d_j, as represented by the blue arrows.

Lemma 6.4. *Let B be a strict barcode, and let $(b_i, d_i), (b_j, d_j)$ be bars of B such that $d_i < d_j$. Let (b'_i, d'_i) and (b'_j, d'_j) denote the corresponding bars in $B' = \mathrm{TMD} \circ \mathrm{TNS}(B)$. The probability that $d'_j < d'_i$ is*

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

The TNS thus exhibits a sort of "transposition stability": the probability that the death times of two bars will be transposed decreases exponentially with the distance between those death times.

Proof. We compute $\mathbb{P}(d'_j < d'_i) = \mathbb{P}(d'_j < d'_i \,|\, d_i < d_j)$, the probability that $d'_j < d'_i$ given that $d_i < d_j$. Denote the random variable $d_i - d'_i$ by X_i. Observe first that

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

Let $Y = X_j - X_i$. As X_j and X_i both follow an exponential law, the density function of their difference, Y, is given by $f_Y(t) = \frac{\lambda}{2} \exp(-\lambda t)$ when $t \geq 0$. Therefore,

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

□

Remark 6.5. Since the TNS is based on a stochastic process, multiple transpositions can occur when generating a new tree from a barcode. This makes it challenging to determine the overall probability of changing equivalence classes when computing the composite TMD $\circ$ TNS. Note that the TNS might also affect the birth order, but we will not discuss this possible effect in this thesis. For the following experiments, the selected examples do not experience birth-switches, as the neurons from which we computed the barcodes were chosen with sufficient gaps between birth values to avoid such switches.

We perform the following computational experiment to evaluate the transposition stability results. We systematically vary the distance between two bars by changing the death time of a bar in the input barcode and compute the percentage of order changes that occur for different values of lambda(see Figure 6.6). We compare the theoretical results (solid lines) to the computational experiment (scatter points) for five different values of lambda. Note that for this experiment, the birth times are chosen to be sufficiently distinct, and only the number of switches due to permutations that correspond to death changes are counted. The experimental results match the theoretical prediction with high accuracy, where we compute the error as the average distance of the computational points from the theoretical curve ($\lambda = 10, error = 0\%$, $\lambda = 5, error = 0.02\%$, $\lambda = 1, error = 0.5\%$, $\lambda = 0.5, error = 0.9\%$, $\lambda = 0.1, error = 3\%$, $\lambda = 0.05, error = 5\%$). Note that the error increases for smaller values of λ, due to the computational artefacts introduced when λ is small.

6.3 Computational Exploration of the TRN

In this section we present computational results that illustrate the complex relationship between the equivalence class of a barcode and its possible tree-realizations.

We first present four results concerning all geometric trees: a computation of the distribution of tree-realization numbers across the set of equivalence classes of strict barcodes for various numbers of bars, a computation of the empirical distribution of combinatorial types of geometric trees in a synthesized population as a function of the equivalence class of the input barcode, a measurement of the diversity of TMD-equivalence classes among the realizations of a fixed barcode, and simulations of the fluctuations in tree-realization number that can occur as two bars gradually switch the order of their deaths.

We conclude by reporting on an experiment that sheds light on the distinguishing characteristics of “biological” geometric trees, i.e., those that arise from digital reconstructions of neurons.

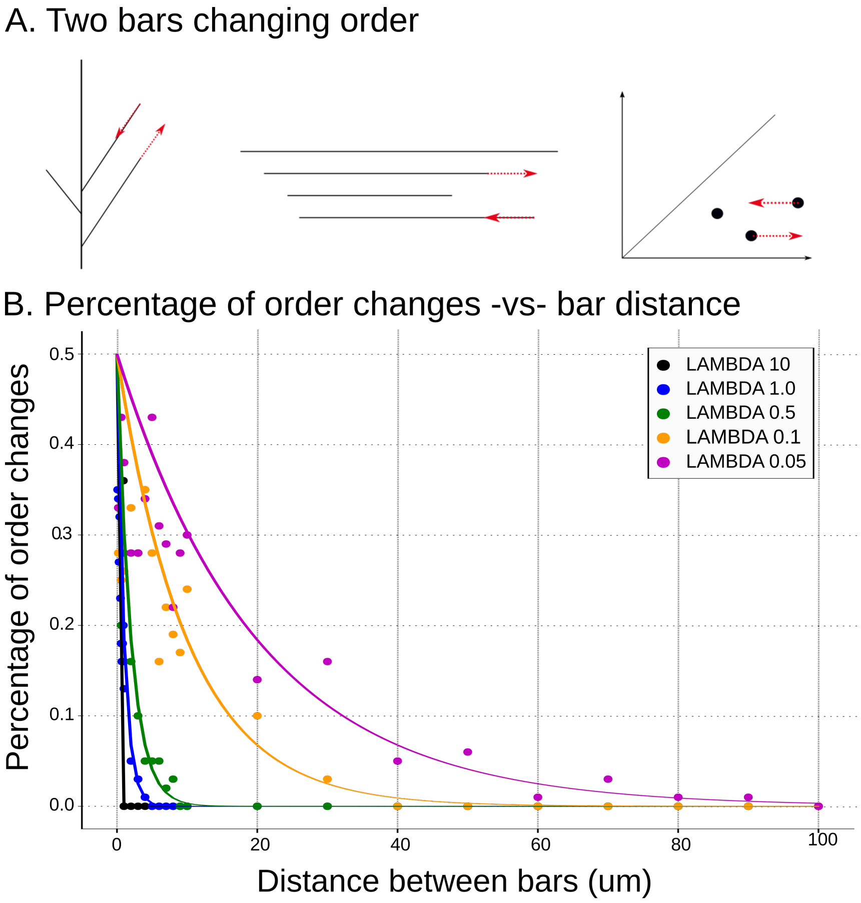


Figure 6.6: A. Example of two bars changing order, which results in switching of classes. We show here a tree, its barcode, and the corresponding persistence diagram when two consecutive deaths switch their order. The impact of the change is illustrated by the red arrows. B. Percentage of order changes per 100 repetitions for varied distance between death times of two consecutive bars of the input barcode. Comparison of theoretical results (solid lines) to simulations (scatter plot) for different values of lambda.

6.3.1 The distribution of tree-realization numbers

We illustrate here how the number of tree-realizations of strict barcodes with $n+1$ bars depends on n. In Figure 6.7 we present the distribution of tree-realization numbers across equivalence classes of barcodes with $n+1$ bars, for $1 \leq n \leq 10$. As mentioned in Chapter 5, the tree-realization number is maximal for a fixed number of bars if and only if the barcode is strictly ordered. We observe an exponential-like behavior in the distribution of tree-realizations with the increase of the number of bars. This illustrates the results of Section 5.3.

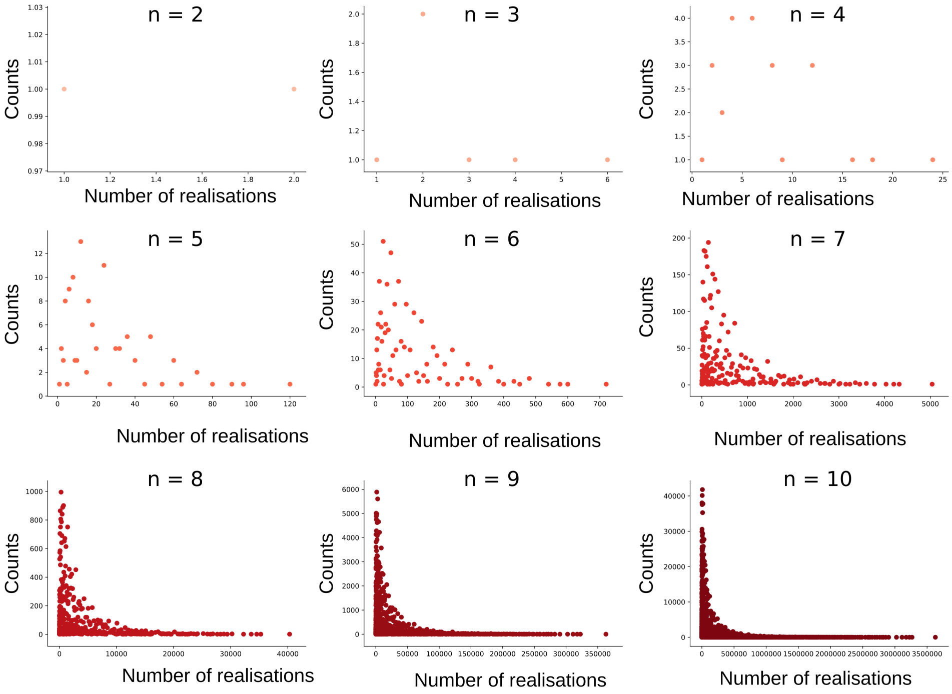


Figure 6.7: Histogram of tree-realization numbers for equivalence classes of barcodes with $n+1$ bars ($1 \leq n \leq 10$). The maximal tree-realization number for a fixed number of bars can be achieved with exactly one equivalence class, that of the strictly ordered Russian doll barcode.

6.3.2 Empirical distributions of combinatorial types of trees

In this section, we explore computationally the probability to generate different combinatorial tree types (see Figure 5.5) with the TNS. We observe that this probability depends on the choice of the parameter λ. When $\lambda > 2$, the TNS is more likely to generate trees with all branches connected to the longest branch, due to the design of the algorithm. On the other hand, for smaller values of λ, the probability to generate different types of trees is approximately uniform.

Focusing on our preferred value of λ, we generated 1000 trees for $\lambda = 1$ and computed the percentage of each combinatorial tree type that is realized for each equivalence class of barcodes with four bars (Figure 6.8). There are six possible equivalence classes of strict barcodes with four bars and six combinatorial equivalence classes of geometric trees with four branches.

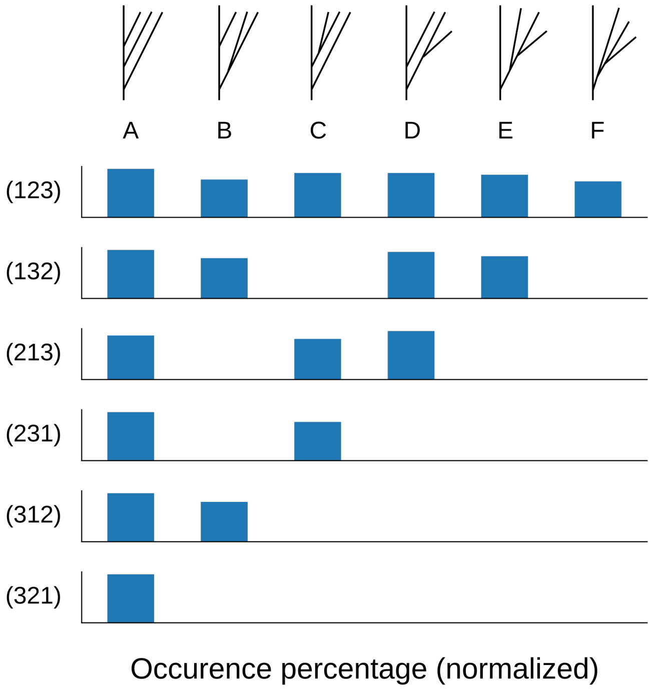


Figure 6.8: Empirical distribution (percentage of 1000 trees) of synthesized geometric trees with four branches by combinatorial tree type (columns A - F) for a given input barcode equivalence class (rows), when $\lambda = 1$. We observe that the distribution is approximately uniform.

6.3.3 Diversity of realized TMD-equivalence classes

Recall that two trees are TMD-equivalent if applying the TMD to both returns the same barcode. We explore the diversity of TMD-equivalence classes of geometric trees that can be synthesized from a fixed barcode, in the particular case of the TMD of a biologically meaningful tree. For a fixed geometric tree with eight branches arising from a digital reconstruction of a layer 4 pyramidal neuron, we computed its TMD barcode, to which we applied the TNS with $\lambda = 1$ to generate a set of 100 geometric trees. We computed the barcode-type and the persistence diagrams of the synthesized trees (Figure 6.9).

In agreement with the results presented in Figure 6.4, the persistence diagrams of the synthesized trees (Figure 6.9B, in blue) are essentially indistinguishable from the persistence diagram of the original barcode (Figure 6.9B, in red). On the other hand, the TMD-equivalence class of a synthesized tree is not necessarily equal to that of the original tree (Figure 6.9A). Here we represent the TMD-equivalence class of a tree in terms of the permutation σ_B corresponding to the equivalence class of its TMD barcode B.

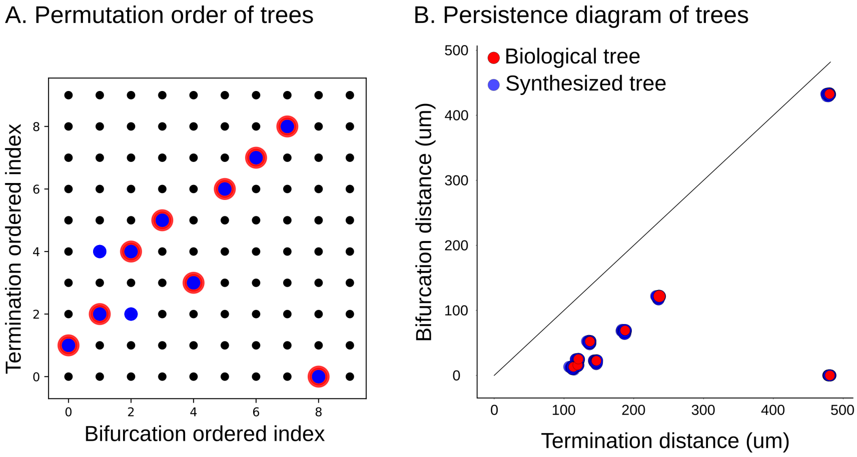


Figure 6.9: Barcode-equivalence class, represented by the corresponding permutation, (A) and persistence diagram (B) of 100 synthesized neurons based on a geometric tree with eight branches, extracted from a layer 4 pyramidal cell. The barcode-equivalence classes of the synthesized trees (represented by blue dots) can differ from that of the original tree due to the stochastic nature of synthesis algorithm. The persistence diagrams of the synthesized trees (B, blue) are essentially indistinguishable from those of the original tree (B, red).

6.3.4 Statistics of changing classes

Motivated by the theoretical results on the probability to change classes in section 6.2.2, we analyze here several simulations of gradual switching of death order of two bars and the resulting effect on tree realizations and their associated barcodes.

Let B be a strict barcode, and let $(b_i, d_i), (b_j, d_j)$ be bars of B such that $d_i < d_j$. By Lemma 6.4, for a fixed choice of the parameter λ, the probability that the order of the death times is reversed in $\mathrm{TMD} \circ \mathrm{TNS}(B)$ depends exponentially on the distance between d_i and d_j:

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

Thus, when the distance between d_i and d_j decreases, the probability that the order of bars changes increases. When there is no k such that $d_i < d_k < d_j$, Proposition 5.7 provides a formula for the tree-realization number of the new barcode obtained when such a switch happens, as long as the order of the birth times is not also reversed.

We start with a geometric tree T extracted from a digital reconstruction of a neuron and compute its associated barcode $B = \mathrm{TMD}(T)$. We choose two bars (b_i, d_i) and (b_j, d_j) of B that are consecutive in the order of deaths and divide the interval (d_i, d_j) into 50 equally sized subintervals. For $0 \leq k \leq 50$, let B_k be a barcode that is identical to B, except that its i^{th} bar is $\big(b_i, d_i + k(d_j - d_i)/50\big)$ and its j^{th} bar is $\big(b_j, d_j - k(d_j - d_i)/50\big)$. An interesting way to visualize this change is to think of B_k as migrating along the edge between B and the barcode with d_i and d_j permuted in the corresponding Cayley graph as k increases. The middle point of the edge corresponds to the non-strict barcode for which the two deaths are equal.

Let $B'_k = \mathrm{TMD} \circ \mathrm{TNS}(B_k)$ for all k. Because of the stochastic nature of the TNS algorithm, the permutation equivalence class of B'_k may be different from that of B_k. Figure 6.10 provides an example of this construction, where the barcodes are represented as persistence diagrams for visualization purposes.

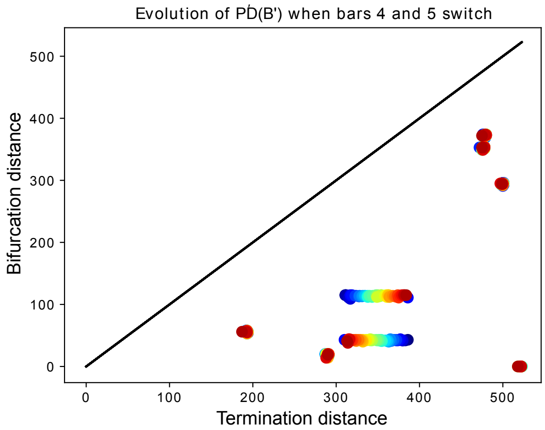


Figure 6.10: We begin with a barcode with 8 bars. The death times d_{i_4} and d_{i_5} (i.e., the 4th and 5th largest death times) are slowly switching as k increases, represented by red-shifting of the color of the points in the persistence diagram. When $k = 0$ (in red), we have the original barcode B, and when $k = 50$ (in blue) we obtain a barcode identical to the original, except that (b_{i_4}, d_{i_4}) is replaced by (b_{i_4}, d_{i_5}) and (b_{i_5}, d_{i_5}) by (b_{i_5}, d_{i_4}).

To test whether the barcode B'_k is equivalent to the original barcode B, we compute its tree-realization number: if $\mathrm{R}(B'_k) \neq \mathrm{R}(B)$, then B and B'_k are not equivalent. Note that for the specific process that gives rise to B_k, it is likely that only the studied death-switch could lead to a difference between the tree-realization numbers of the input and output barcodes, unless two other deaths are too close to each other in the input barcode, as in the last row of Figure 6.11, which we explain further below. Therefore, the tree-realization number provides a very good indication of whether the switch of deaths took place, i.e., if $B \sim B'_k$ or not. Indeed, two barcodes that are the same except for two deaths that switched have different tree-realization number, cf. Proposition 5.7. Figure 6.11 shows several examples of the endpoint-switching process described above and the corresponding evolution of the tree-realization number as k increases. The corresponding permutation type and tree-realization number of each initial barcode, and the bars that are switched are listed in Table 6.2.

The top row of Figure 6.11 illustrates very well the exponential behavior of changing classes. When the distance between the death times of the two bars is very small (they are the closest when $k = 25$), the tree-realization number oscillates between its values for two different classes and otherwise stays constant.

The two middle rows come from the same biological tree and hence have the same starting barcode. The difference is that in the second row, the death times of the two bars are already very close, leading to more frequent changes of equivalence class than in the third row.

The bottom row illustrates Remark 6.5 well. Since several bars are close to each other (represented here by several points in the persistence diagram that are close to each other), applying the TNS algorithm leads to frequent changes in equivalence classes, leading to the oscillatory behavior of the tree-realization number curve.

	Permutation	**TRN**	**Bars that switch**
B^1	$[2, 6, 8, 1, 5, 7, 4, 3]$	810	4 and 5
$\hat{B}^1$	$[2, 6, 8, 5, 1, 7, 4, 3]$	540	4 and 5
B^2	$[5, 7, 6, 4, 2, 1, 3]$	12	2 and 3
$\hat{B}^2$	$[5, 6, 7, 4, 2, 1, 3]$	18	2 and 3
B^3	$[5, 7, 6, 4, 2, 1, 3]$	12	3 and 4
$\hat{B}^3$	$[5, 7, 4, 6, 2, 1, 3]$	18	3 and 4
B^4	$[8, 6, 7, 4, 3, 1, 2, 5]$	20	1 and 2
$\hat{B}^4$	$[6, 8, 7, 4, 3, 1, 2, 5]$	40	1 and 2

Table 6.2: For each example displayed in Figure 6.11, we list the permutation type and the tree-realization number of the original barcode B and of $\hat{B} = B_{50}$, and the indices of the bars that are switched. The superscript i in B^i indicates the corresponding row of Figure 6.11. For example, the largest death time of barcode B^1 is the second bar (in order of birth times), and its shortest death is the third one. When we switch the 4th and 5th (from largest to smallest) death times in B^1 and $\hat{B}^1$, the TRN changes from 810 to 540.

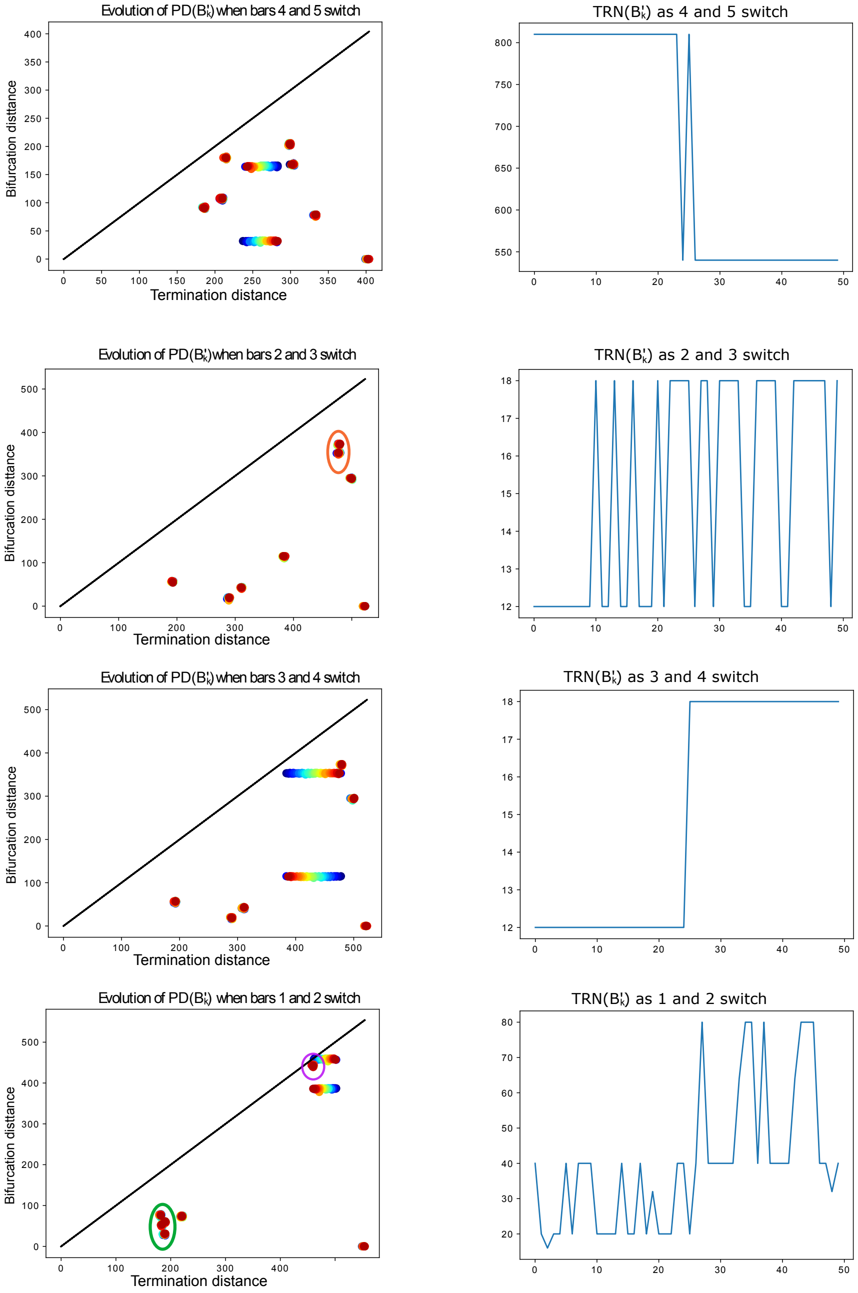


Figure 6.11: On the left, evolution of $\mathrm{PD}(B'_k)$ as k increases (represented by red-shifting of the point color, from red $k = 0$ to blue $k = 50$), for various pairs of bars. When not clear, we circle in orange the two points that switch. On the right, the corresponding evolution of the tree realization number $\mathrm{R}(B'_k)$ as k increases. For instance, as indicated in Table 6.2, the tree-realization number of B^1 is 810 and that of $\hat{B}^1 = B^1_{50}$ is 540. The barcodes B'_k exhibit the behavior described in Lemma 6.4, except for the last row, in which death times that are too close to each other (circled in purple and green) interfere with the process. Without this interference, the tree-realization numbers should oscillate between 20 and 40. When k gets close to 50 (blue), the death time d_{i_1} (largest death time) starts interfering with the third one d_{i_3} (circled in purple) in the tree synthesis process.

6.3.5 Tree-realizations of biological barcodes

Since the original objective in developing the TMD was to classify digital reconstructions of neurons, it is natural to ask whether those barcodes that arise biologically exhibit any special characteristics compared to those arising from other sets of geometric trees. In Figure 6.12 we employ the graphical representation of permutations introduced in Section 6.3.3 to display as red dots all possible permutations corresponding to TMD-barcodes of biological trees with at most 30 branches arising from a population of digital reconstructions of neurons. Clearly, only a small fraction of the set of all possible permutations can be realized as the barcode-equivalence classes of geometric trees extracted from digital reconstructions of neurons, as every black dot in this plot can arise as a pair $\big(k, \sigma(k)\big)$ for some permutation σ.

To provide further insight into the subset of TMD-equivalence classes of biological geometric trees within the set of all possible TMD-equivalence classes, we computed the tree-realization number as a function of the number of bars, for a population of barcodes obtained by applying the TMD to geometric trees extracted from a population of digitally reconstructed neurons. We compared the values obtained to the maximum tree-realization number and to the tree-realization numbers of randomly chosen barcodes with the same number of bars (Figure 6.13). This is a similar study that was shown in the introduction, Figure 1.2. Interestingly, the barcodes that correspond to apical dendrites (relatively complex neural trees that perform significant processing tasks) exhibit a more narrow range of possible tree-realization numbers than random barcodes of the same size. On the other hand, barcodes of basal dendrites (less complex neuronal trees) exhibit tree-realization numbers similar to those of the randomly generated barcodes.

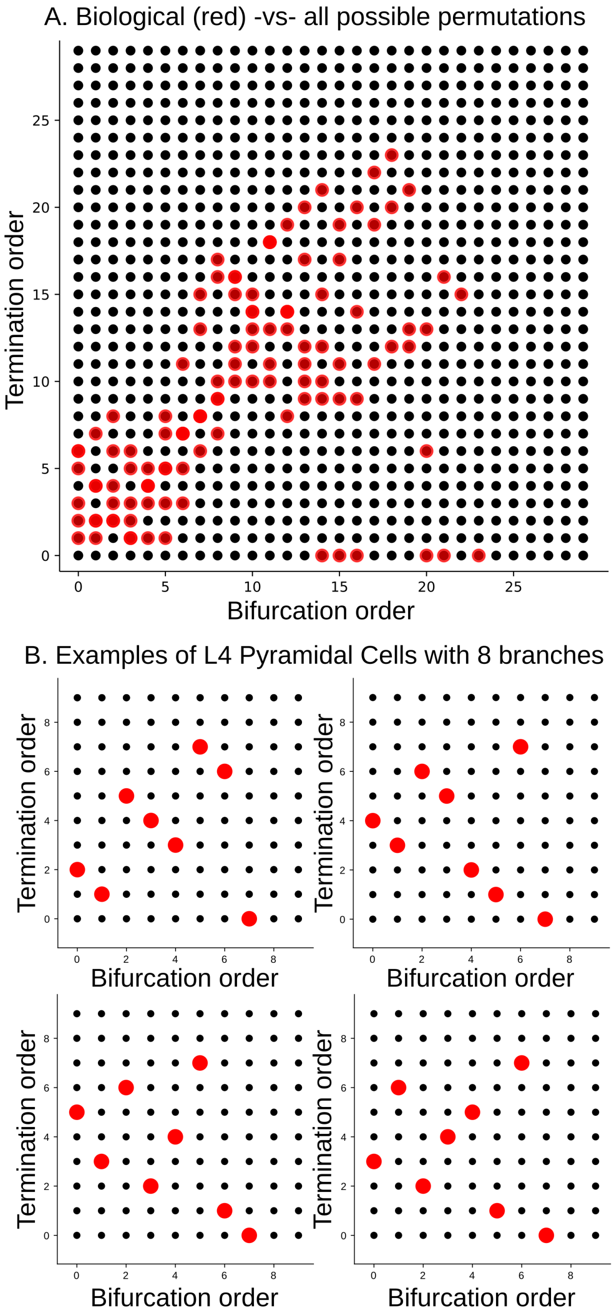


Figure 6.12: (A) TMD-equivalence classes of a population of biological geometric trees with at most 30 bars (red dots), represented by their associated permutations. (B) Examples of TMD-equivalence classes of individual biological geometric trees with eight branches, extracted from layer 4 pyramidal neurons (red dots).

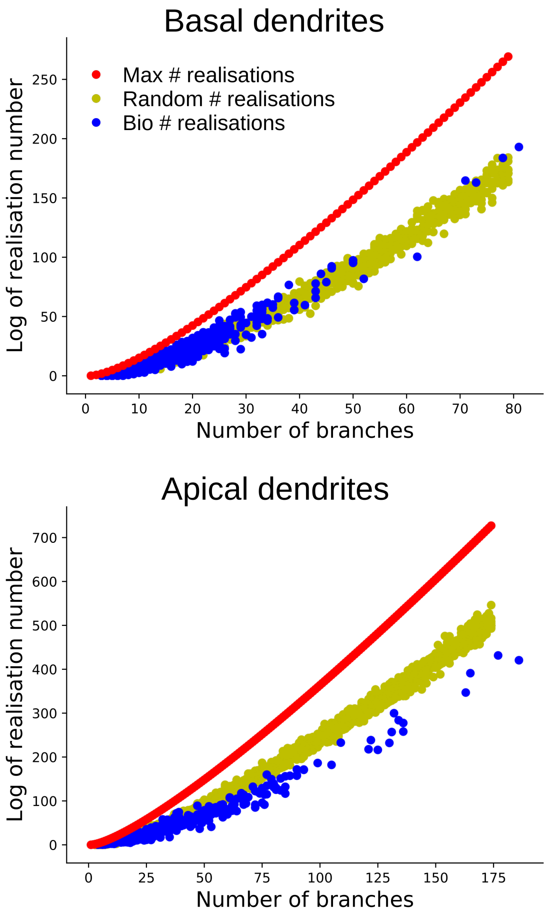


Figure 6.13: The log of the tree-realization number for barcodes with varying numbers of bars. (A) The log of tree-realization number for barcodes of basal dendrites (in blue) in comparison with random barcodes (in yellow) and the maximum tree-realization number ($n!$ for $n + 1$ bars) (in red). (B) The log of the tree-realization number for barcodes of apical dendrites (in blue) in comparison with random barcodes (in yellow) and the maximum maximum tree-realization number (in red).

6.3.6 Comparison with Random Barcodes

Motivated by the results in Figure 6.13, we go further in the comparison of distributions of barcodes coming from neurons and artificially generated barcodes. The computations in this section were done by two interns at EPFL, Jeanne Fernandez and Ettore Gran.

For normalization purposes, all the methods to generate barcodes take values in $[0, 100]^2$. The barcodes of neurons are computed computed from publicly available data on NeuroMorpho.org [6]. The four data set we study in this section are fly neurons [30], mouse basal glanglia cells [97], mouse neocortex cells [97] and rat neurons [80]). We use the TMD [68] to compute the barcodes, which we normalize by dividing the endpoint values in each barcode by its maximal death time. We compare these biological barcodes with artificially generated barcodes.

We describe several methods to generate artificial barcodes below.

- To generate a barcode with n bars, the first method (m1) repeats n times the following procedure: pick $b_i \in [0, 100)$, then pick $d_i \in (b_i, 100]$. Because the latter distribution is conditioned on $d_i > b_i$, the induced distribution on the symmetric group is not uniform, as seen in Figure 6.14 (green).
- The second method (m2), picks two values $x_i, y_i \in [0, 100]$ and defines $b_i = \min\{x_i, y_i\}$ and $d_i = \max\{x_i, y_i\}$. This method also does not induce a uniform distribution on the symmetric group. Indeed, the distribution of the minimum or maximum of two uniform random variables is not uniform.
- The third method (m3) is designed to induce a uniform distribution on the symmetric group. The birth b_i is always given the value i and the death values are picked uniformely in $(n, 100]$.
- The fourth method (m4) is similar to (m1), but starts with the death values. It picks $d_i \in (0, 100]$ and then conditions $b_i \in [0, d_i)$. It is used in Figure 1.2 (green curve) and Figure 6.13 (yellow curve) to study the distribution of the tree-realization number for populations of neurons.
- Another method (separate) designed to induce a uniform distribution on the symmetric group is to pick the birth and death times seperately: pick $b_i \in [0, 50)$ and $d_i \in (50, 100]$. Figure 6.14 shows the distribution induced on Sym_3 by this method and the method (m1). This method is used in Figure 1.2 to compare the tree-realization number of random barcodes and neurons. It corresponds to the blue curve.
- The last method (plane) that we use picks a point randomly in the region of the plane defined by $[0, 100]^2 \cap \{(x, y) \in \mathbb{R}^2 \mid x < y\}$. To do so, one generates polar coordinates with an angle between $\frac{\pi}{4}$ and $\frac{\pi}{2}$ and a radius in $[0, 100]$.

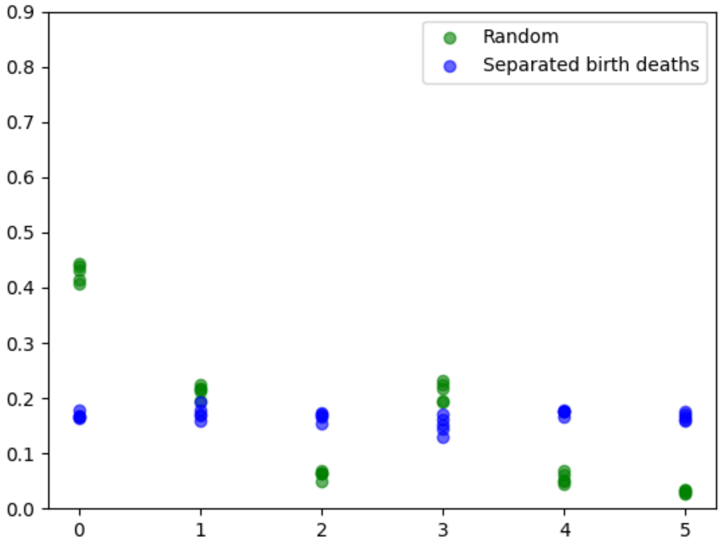
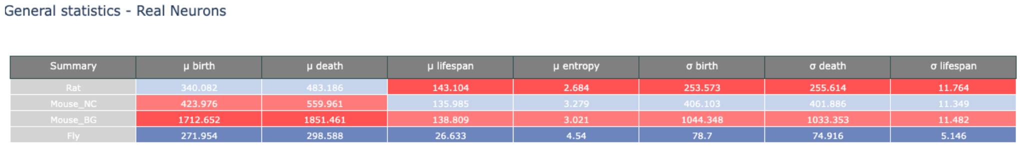


Figure 6.14: Two distributions on Sym_3 induced by distributions of barcodes with three non-essential bars. The elements of Sym_3 are indexed by the integers $0, ..., 5$. Green (m1): we first pick uniformly the birth times b_i in the interval $[0, 100]$, then choose uniformly the death times $d_i \in [b_i, 100]$. Blue (separate): We pick uniformly three birth times $b_i \in [0, 49]$ and three death times $d_i \in [50, 100]$, which induces a uniform distribution on the symmetric group S_3.

In the following figures, we compare populations of neurons to artificially generated barcodes using different statistics. To generate barcodes with the methods described above, we first studied the distribution of the number of bars of each population of neurons. We mimic these distribution for our artificially generated barcodes by generating five barcodes with n bars for each barcode with n bars in the population of neurons.

We first show a summary of the data of each population of neurons of rats, mice (neocortex and basal ganglia cells) and flies in Figure 6.15.

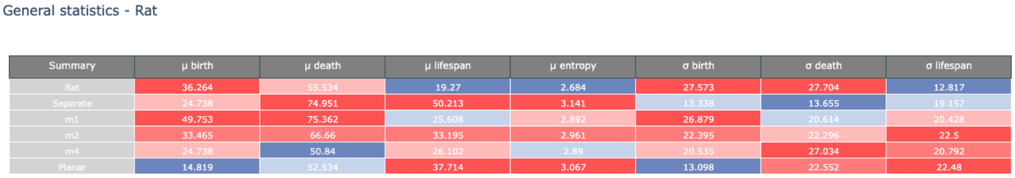


General statistics - Real Neurons

Summary	μ birth	μ death	μ lifespan	μ entropy	σ birth	σ death	σ lifespan
Rat	340.082	483.186	143.104	2.684	253.573	255.614	11.764
Mouse_NC	423.976	559.961	135.985	3.279	406.103	401.886	11.349
Mouse_BG	1712.652	1851.461	138.809	3.021	1044.348	1033.353	11.482
Fly	271.954	298.588	26.633	4.54	78.7	74.916	5.146

Figure 6.15: Table summarizing the normalized births, deaths, lifespans $(d_i - b_i)$ and entropy averages (μ) and standard deviations (σ) for the fly data set, mouse basal ganglia (mouse_BG), mouse neocortex mouse_NC) and rat data set.

The next table 6.16 shows the same summaries for the normalized barcodes of rat neurons. They are compared with the same summaries for the methods described above to generate artificial barcodes.



General statistics - Rat

Summary	μ birth	μ death	μ lifespan	μ entropy	σ birth	σ death	σ lifespan
Rat	36.264	55.534	19.27	2.684	27.573	27.704	12.817
Separate	24.738	74.951	50.213	3.141	13.338	13.655	19.157
m1	49.753	75.362	25.608	2.892	26.879	20.614	20.428
m2	33.465	66.66	33.195	2.961	22.395	22.296	22.5
m4	24.738	50.84	26.102	2.89	20.535	27.034	20.792
Planar	14.819	52.534	37.714	3.067	13.098	22.552	22.48

Figure 6.16: Table summarizing the normalized births, deaths, lifespans $(d_i - b_i)$ and entropy averages (μ) and standard deviations (σ) for the rat data set compared to randomly generated barcodes.

Figures 6.17, 6.18, 6.19 and 6.20 show box plots of the births and deaths for each population of barcodes of neurons compared to each method described above. For each plot, the values are represented with dots, the median, upper and lower quartiles with boxes and the upper and lower extremes with whiskers. For artificial barcodes, the birth and death boxes are in the same color, the left box corresponds to the births and the right one to the death, since $b_i < d_i$ for all i. These plots show that the methods to generate artificial barcodes are not a good approximation of the neurons' barcodes. This is not surprising due to the simplicity of these methods.

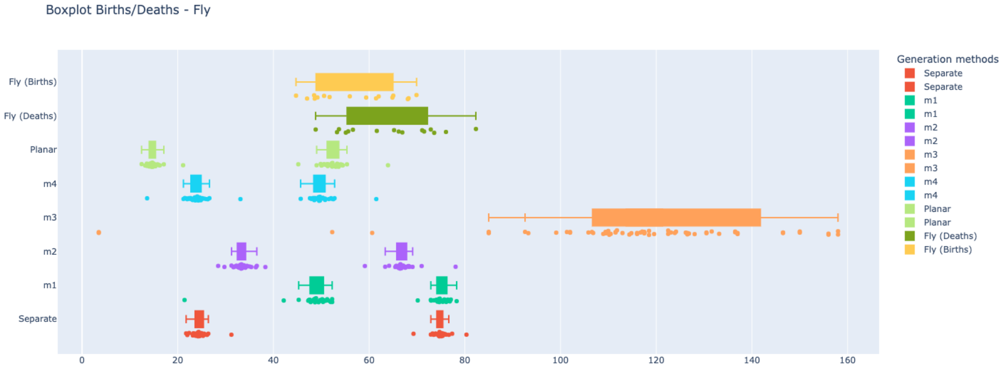


Figure 6.17: Box plot of the birth and death values for the normalized barcodes of the fly data set and the methods to generate artificial barcodes.

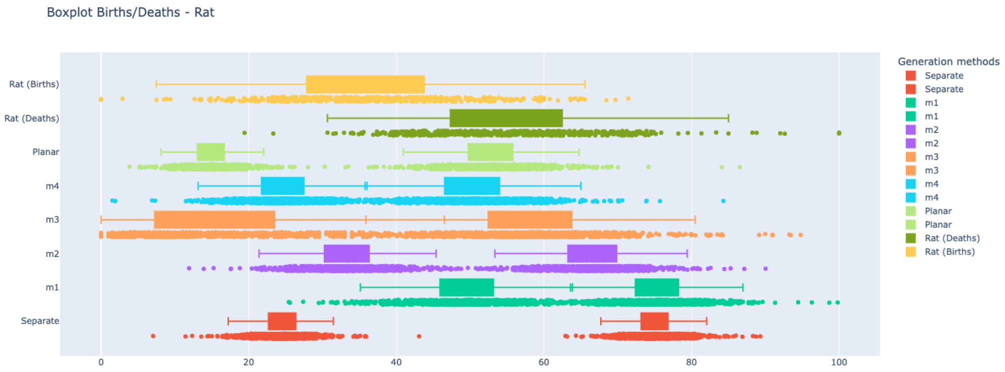


Figure 6.18: Box plot of the birth and death values for the normalized barcodes of the rat data set and the methods to generate artificial barcodes.

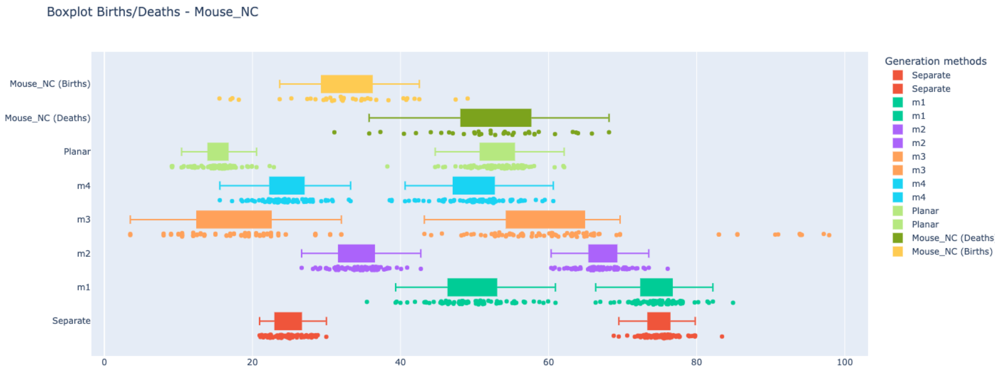


Figure 6.19: Box plot of the birth and death values for the normalized barcodes of the mouse basal ganglia data set and the methods to generate artificial barcodes.

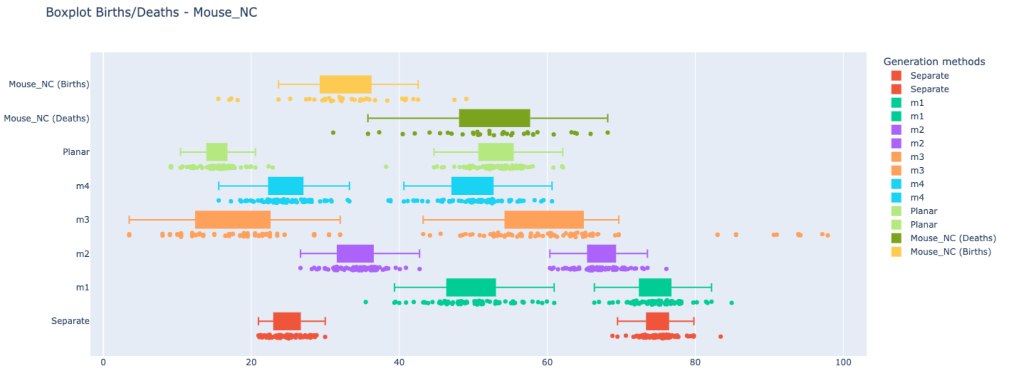


Figure 6.20: Box plot of the birth and death values for the normalized barcodes of the mouse neocortex data set and the methods to generate artificial barcodes.

We now compare the (log of) tree-realization number with the persistent entropy (Theorem 3.29) of the different neuron types and then compare the rat and mouse basal ganglia data sets with the different generation methods. The plots ressemble those for the TRN and the number of bars of Figures 1.2 and 6.9. Both the TRN and persistent entropy are correlated with the number of bars.

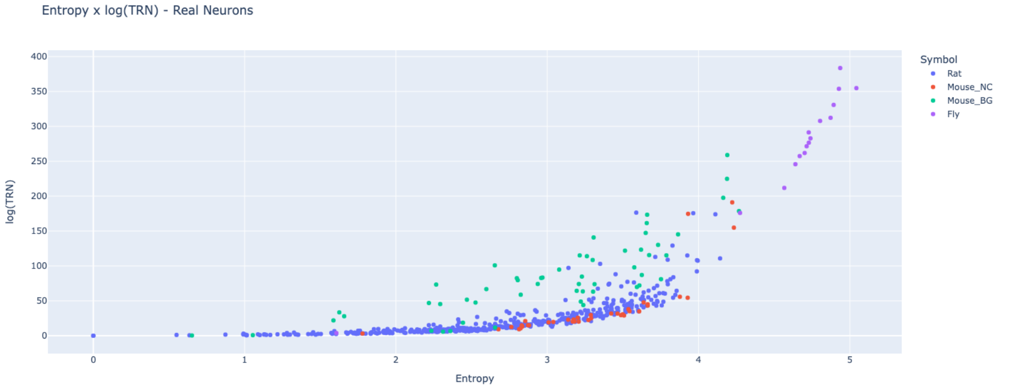


Figure 6.21: Log of the tree-realization number versus entropy of the barcodes for the neurons data set.

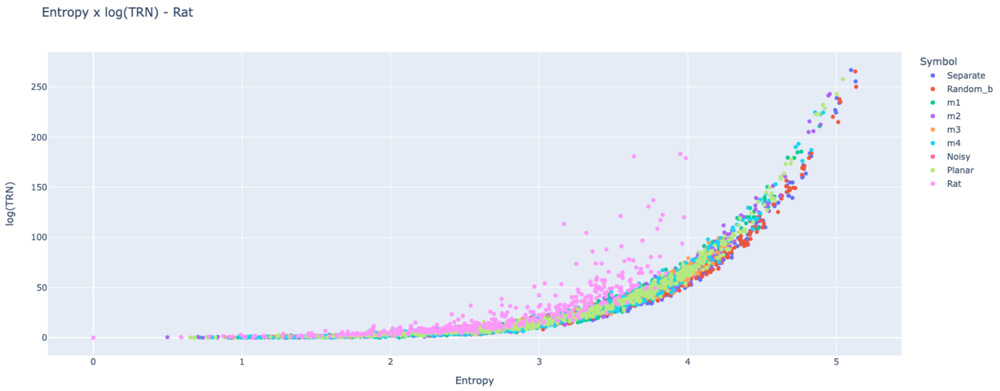


Figure 6.22: Log of the tree-realization number versus entropy of the barcodes for the rat data set compared with artificially generated barcodes.

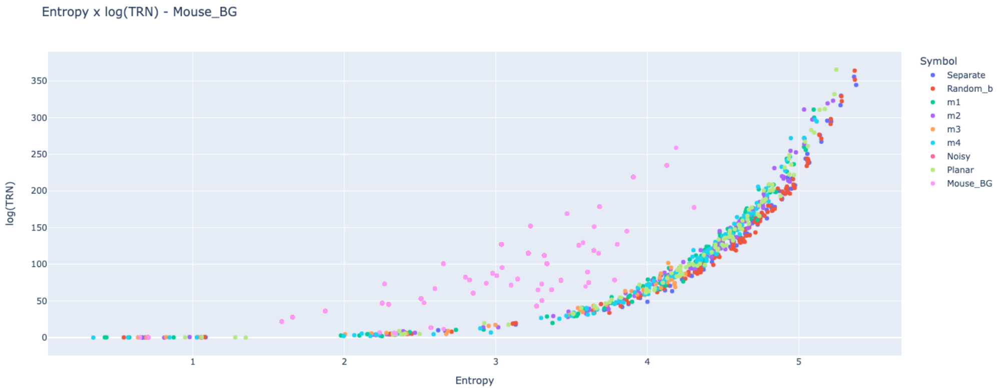
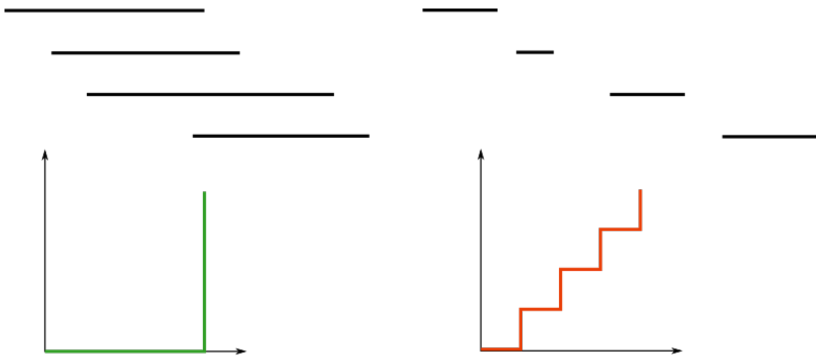


Figure 6.23: Log of the tree-realization number versus entropy of the barcodes for the mouse basal ganglia data set compared with artificially generated barcodes.

Another combinatorial statistical study

While the permutation type of barcode and tree-realization number have been shown to have interesting properties with respect to the inverse problem, they yield little infomation about the “overlapping” properties of births and deaths. Indeed, a barcode that has the identity permutation can have alternating birth and deaths times, as in Figure 6.24A, or all the births separated from the deaths, as in Figure 6.24B. There is a family of possible “overlapping behavior” in the middle. The Russian doll barcode, on the other hand, can only have all births separated from the death times.

Let $B = \{(b_i, d_i)\}_{i \in \{0,\ldots,n\}} \in \mathcal{B}_n$. In order to understand the overlapping behavior of the births and deaths, we order all the values in a labelled list $(x, y) \in \mathbb{R}^{2n} \times \{b, d\}^{2n}$. The value $x_i \in \{b_j\}_{j \in \{0,\ldots,n\}} \cup \{d_j\}_{j \in \{0,\ldots,n\}}$ is either a birth or a death, $x_i < x_j$ for $i < j$, and $y_i \in \{b, d\}$ is a binary variable indicating whether x_i is a birth or a death. Keeping only the second coordinates y_i yields an ordered list in $\{b, d\}^{2n}$ that summarizes the "alternating behavior" of the births and deaths. We represent this information graphically in what we call a *path diagram*. The x-axis represents the births and the y-axis the deaths. We draw a path that goes right if y_i indicates a birth and up if y_i indicates a death. Therefore, a path close to the diagonal indicates an alternating behavior such as (b, d, b, d, b, d, b, d) (Figure 6.24B) and a path that first follows the x-axis then goes up would indicate a separate birth/death behavior such as $(b, b, b, b, b, d, d, d, d, d)$, as in Figure 6.24A.



A. A barcode with (b,b,b,b,d,d,d,d) and its corresponding path diagram.

B. A barcode with (b,d,b,d,b,d,b,d) and its corresponding path diagram.

Figure 6.24: The two extreme types of paths. (A) A barcode with all births separated from the deaths (b, b, b, b, d, d, d, d) and its corresponding path diagram. The path goes first 4 times to the right for all the births, then 4 times up for the deaths. (B) A barcode with alternating births and deaths (b, d, b, d, b, d, b, d). The corresponding path goes right, up, right, up, right, up, right, up.

The plot of Figure 6.25 shows the path diagrams for several neurons of the rat data set chosen for their low number of bars.

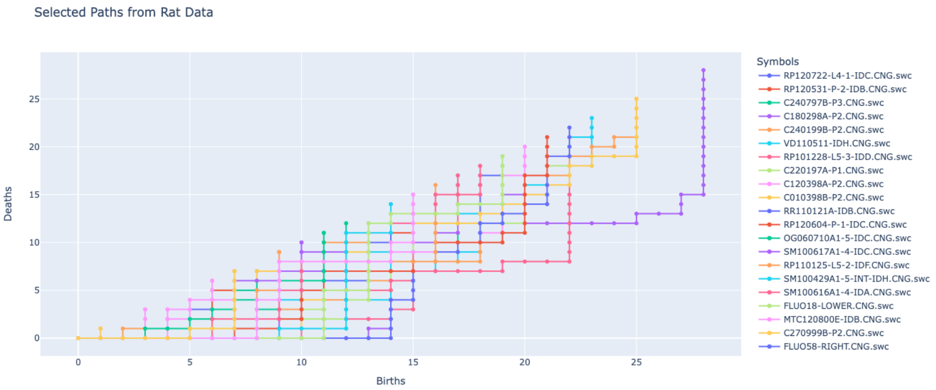


Figure 6.25: The path diagrams of several neurons from the rat data set.

The heat map corresponding to Figure 6.25 is shown in Figure 6.26. It shows the coordinates $(x, y) \in \mathbb{R}^2$ which are contained in the most number of paths (yellow).

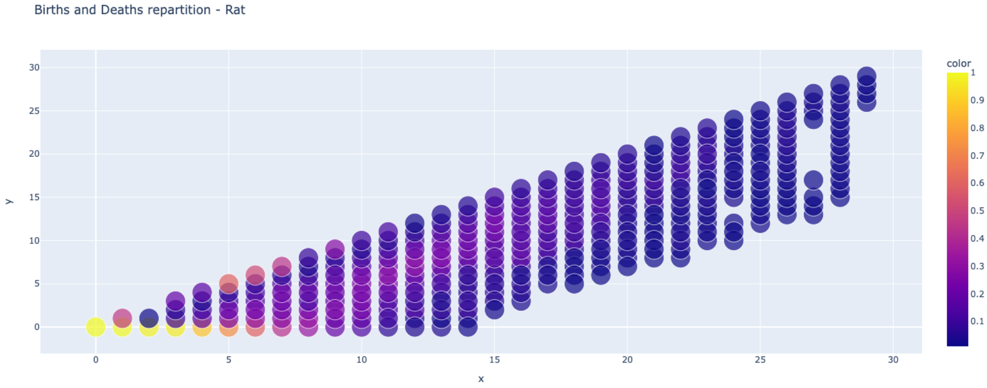


Figure 6.26: Heat map corresponding to Figure 6.25.

Based on Figures 6.25 and 6.12, one can formulate the following hypothesis for (rat) neuron's barcodes. They have the tendency to have permutations that are close to the identity (see Figure 6.12) but still have a rather alternating behavior of birth and death times (see Figure 6.25). Such an hypothesis is formulated purely from the combinatorics of the barcodes, using tools developed in this thesis. We discuss future work related to combinatorial properties of barcodes and trees in the conclusion.

Chapter 7

The Persistent Homology of Dual Filtered Complexes

The purpose of this chapter is to present results on the persistent homology of dual cell complexes. It is self-contained (following the definitions of Sections 2.4 and 7.1) and independent of the other results of this thesis. The results presented are based on two papers [16, 58], and are joint work with K. Maggs, V. Robins, T. Heiss and B. Bleile. In [16], we apply these results to the computation of persistent homology for the two types of cubical complexes built from images.

In this chapter, a cell complex is a finite regular CW complex or a simplicial or cubical complex. Given a d-dimensional complex X, recall that, when it exists, its dual X^* is a complex whose k-dimensional cells correspond bijectively to the $(d-k)$-dimensional cells of X, and the adjacency relations are reversed. It was observed in [27], [9] that a Morse function on a manifold satisfies some duality relations. Here, we show that the persistent homology of two dual cell complexes satisfies similar relations. These relations can be lifted to the chain complex level, by exhibiting a shifted filtered chain isomorphism between the absolute filtered cochain complex of X and the relative filtered chain complex of X^*. This induces a *natural* isomorphism between the absolute persistent cohomology of X and the relative persistent homology of X^*.

In [37], it is shown that the persistent homology and cohomology of a space have isomorphic barcodes, and that there is a bijection between the relative persistent homology and absolute persistent homology barcodes. Using these bijections, we extend our results to a bijection between the (absolute persistent homology) barcodes of X and X^*. Moreover, we formalize the notion of dual in the context of discrete Morse theory. The filtered chain isomorphism described above is on the Morse chain complexes, generalising results of [27] to cell complexes.

We start by giving a brief introduction to discrete Morse theory.

7.1 Background on Discrete Morse Theory

Discrete Morse theory is a version of Morse theory for discrete structures such as simplicial or cubical complexes. The notion of a discrete Morse function on a cell complex generalizes that of a Morse function on a manifold. Discrete Morse theory, as for Morse theory, studies the critical cells of a complex K. The homology (and persistent homology) of K can be computed directly from the critical cells and the discrete gradient vector field, illustrating the power of (discrete) Morse theory.

From a discrete Morse function, one can build an algebraic *Morse chain complex* and compute its homology. This construction can be found in detail in [52]. If one has a filtered cell complex and a discrete Morse function on it, one can also compute its persistent homology, which was shown in [88] to correspond to the persistent homology of the underlying filtration.

7.1.1 Discrete Gradient Vector Fields

Let K be a finite cell complex whose cells will be denoted by α or β,with superscripts denoting the dimension of the cell. A pair of cells (α, β) is called a *free pair* if $\alpha < \beta$ (here, $<$ denotes "is a proper face of") and α has no other coface in K. One calls $K \searrow \{\alpha, \beta\}$ an *elementary collapse* of K. A function $f : K \longrightarrow \mathbb{R}$ is a *discrete Morse function* if for all $\alpha \in K$:

1. $\mid \{\alpha^{(p)} \lhd \beta^{(p+1)} \mid f(\alpha) \geq f(\beta)\} \mid \leq 1$, and
2. $\mid \{\alpha^{(p)} \rhd \beta^{(p-1)} \mid f(\alpha) \leq f(\beta)\} \mid \leq 1$.

If both sets have cardinality 0 for some α, then α is called a *critical cell* of f. The set of critical cells is denoted by $\mathsf{Crit}(f)$.

The main idea of discrete Morse theory is to define a *discrete gradient vector field* (DGVF) on a complex, which can be used to simplify the complex without changing its homotopy type. A DGVF is a discrete version of a gradient vector field in differential geometry. A discrete vector field V is a partition of the cells into singletons or pairs such that each pair consists of a cell and one of its faces of co-dimension 1.

Definition 7.1. [52] Let X be a cell complex. A *discrete vector field* V is a collection of facet/cofacet pairs $\{(\tau_\lambda^k \lhd \sigma_\lambda^{k+1})\}_\lambda$ such that each cell belongs to at most one pair. Cells that are not paired are called *critical*, and the set of critical cells is denoted by $\mathsf{Crit}(V)$.

A pair $(\tau_\lambda^k \lhd \sigma_\lambda^{k+1})$ in V can be represented visually as an arrow going from τ to σ. The condition of having each cell in at most one pair implies that each cell of X either is the head or the tail of an arrow or is critical.

Flow-lines of a Morse function in the smooth category have a discrete analogue in V-paths, where we piece together pairs of cells in the discrete vector field.

Definition 7.2. [52] Given a discrete vector field V, a k-dimensional *V-path* is a sequence of cells

$$(\tau_0 \lhd \sigma_0 \rhd \tau_1 \lhd \sigma_2 \rhd \ldots \rhd \tau_n)$$

such that for all $0 \leq i \leq n$, we have $(\tau_i \lhd \sigma_i) \in V$, $\tau_i \neq \tau_{i-1}$ and $\dim(\tau_i) = k$.

A *discrete gradient vector field* is a vector field that does not admit any closed V-path, i.e., there is no V-path of which the first and last cells are the same, $\tau_0 = \tau_n$.

If two cells τ and σ are critical, we say that a V-path γ *goes from* σ to τ if γ starts at a cell in the boundary of σ and ends with τ.

Remark 7.3. Given a discrete gradient vector field V, one can always build a discrete Morse function f such that $(\tau \lhd \sigma) \in V$ implies $f(\tau) \geq f(\sigma)$ and $(\tau \lhd \sigma) \notin V$ implies $f(\tau) < f(\sigma)$ [52]. Therefore, $\mathrm{Crit}(f) = \mathrm{Crit}(V)$. Conversely, given a discrete Morse function f, there is a unique discrete gradient vector field V such that $\mathrm{Crit}(f) = \mathrm{Crit}(V)$.

The collapse theorem (Theorem 6.3 and 6.4 in [52]) states that a cell complex X equipped with a DGVF V is homotopy equivalent to a complex composed of the critical cells of V, via a series of elementary collapses.

A discrete gradient vector field pairs cells in a cell complex X. However, if X is filtered, it might happen that two cells that are paired do not appear at the same time in the filtration. Therefore, we add a coherence condition for vector fields in the filtered case.

Definition 7.4. Given a filtration of cell complexes $\{ X_i \}_{i \in I}$, a *filtered discrete gradient vector field*, or *filtered vector field* for short, is a discrete gradient vector field V on X such that for each pairing of cells $\tau \lhd \sigma$, $\sigma \in X_i$ if and only if $\tau \in X_i$.

Since each pairing occurs only at a single filtration step, one can compute the persistent homology of the filtration using only the critical cells of X and the DGVF, as can be done for the homology of X. It was shown in [83] that this computation produces the usual persistent homology. In particular, only critical cells induce a change in (persistent) homology.

7.1.2 Morse Chain Complex

One of the purposes of discrete Morse theory is to simplify computations. Theorem 2.5 in [52] states that for a cell complex X together with a discrete Morse function

f, there exists a cell complex X' homotopy equivalent to X consisting of exactly one cell of dimension p for each critical cell of dimension p in X.
The information about critical cells can be packaged together into an algebraic chain complex, which computes the homology of the space, significantly reducing the computations.

Let V be a discrete gradient vector field over a cell complex X, and let $\Gamma(\tau, \sigma)$ denote the set of V-paths from τ to σ. Here, we work only with $\mathbb{F}_2$-coefficients, so we do not need to define orientation of V-paths. However, the interested reader can consult [52] to see how the following definitions can be extended to $\mathbb{Z}$-coefficients.

Definition 7.5. [52] Let X be a cell complex and V a DGVF on X. The *algebraic Morse complex* $\mathcal{M}$ of (X, V) is a chain complex given by the following data:

1. The chain groups

$$C_n(\mathcal{M}; \mathbb{F}_2) := \bigoplus_{\sigma^{(n)} \in \mathsf{Crit}(V)} \mathbb{F}_2 \cdot \sigma^{(n)}$$

2. The boundary operators $\partial_n^{\mathcal{M}} : C_n(\mathcal{M}; \mathbb{F}_2) \to C_{n-1}(\mathcal{M}; \mathbb{F}_2)$ given by

$$\partial_n^{\mathcal{M}}(\sigma^{(n)}) = \sum_{\nu^{(n-1)} \in \mathsf{Crit}(V)} [\sigma : \nu] \nu^{(n-1)},$$

where $[\sigma : \nu]$ is the number of V-paths from σ to ν modulo 2.

The n-th homology of $\mathcal{M}$ is computed via $\mathrm{Ker}(\partial_n^{\mathcal{M}}) / \mathrm{Im}(\partial_{n+1}^{\mathcal{M}})$, which we denote by $H_n(\mathcal{M}; \mathbb{F}_2)$.

The homology of the Morse chain complex is isomorphic to the singular homology of X.

Theorem 7.6. *[52] (Discrete Morse Homology Theorem)*

$$H_n(\mathcal{M}; \mathbb{F}_2) \cong H_n(X; \mathbb{F}_2).$$

Moreover, if the DGVF is filtered, an analogue of Theorem 7.6 is true for *persistent* homology using the filtered chain complexes [88]. Therefore, the filtered Morse chain complex can be used to compute the persistent homology of a filtered cell complex, which we take advantage of in the last section of this chapter.

7.2 Persistent Homology of Dual Filtrations

Recall again that in singular homology and cohomology with field coefficients, the coboundary map is isomorphic to the adjoint of the boundary map. In particular, given a consistent choice of bases for the chain and cochain groups, their matrix representations are transpose of each other.

In [37], another algebraic relationship is established between persistent homology and persistent relative cohomology, based on the observation that the filtration for relative cohomology reverses the ordering of cells in the total (co)boundary matrix. The same reversal of ordering holds for the dual filtered cell complexes defined here, so we obtain a similar relationship between the persistence diagrams. Our proof of the correspondence between persistence pairs in dual filtrations uses the matrix rank function and pairing uniqueness lemma in a way similar to the combinatorial Helmoltz-Hodge decomposition of [47].

Suppose (X, f) and (X^*, g) are dual filtered cell complexes with $n+1$ cells. Suppose that a linear ordering $\sigma_0, \sigma_1, \dots, \sigma_n$ of the cells in X is compatible with the filtration (X, f), and that $\sigma_n^*, \sigma_{n-1}^*, \dots, \sigma_0^*$ is the dual linear ordering compatible with g. Let D be the total boundary matrix of X and D^* the total boundary matrix of X^*, with respect to their respective orderings.

Remark 7.7. A useful indexing observation is that σ_i^* is the $(n-i)$-th cell of the dual filtration.

We denote by $D^\perp$ the anti-transpose of the matrix D, that is the reflection across the minor diagonal: $D^\perp_{i,j} = D_{n-j,n-i}$. Anti-transposition is also the composition of standard matrix transposition with a reversal of the order of the columns and of the rows.

Lemma 7.8. *The matrix D^* is the anti-transpose $D^\perp$ of D, that is,*

$$D^*_{i,j} = D^\perp_{i,j} = D_{n-j,n-i} \textit{ for all } i, j.$$

Proof. The equivalences below follow from the definition of D, of dual cell complexes, and the remark above.

$$D_{n-j,n-i} = 1 \Leftrightarrow \sigma_{n-j} \lhd \sigma_{n-i} \Leftrightarrow \sigma^*_{n-i} \lhd \sigma^*_{n-j} \Leftrightarrow D^*_{i,j} = 1.$$

□

Lemma 7.9. *The sub-matrices defined in Section 2.4.3 satisfy*

$$(D_i^j)^\perp = (D^\perp)_{n-j}^{n-i}$$

and thus

$$\operatorname{rank} D_i^j = \operatorname{rank} (D^\perp)_{n-j}^{n-i}$$

and

$$r_D(i,j) = r_{D^\perp}(n-j, n-i).$$

Proof. The first statement follows from

$$(D_i^j)^\perp = (D[i:n, 0:j])^\perp = D^\perp[(n-j):n, 0:(n-i)] = (D^\perp)_{n-j}^{n-i}.$$

The second statement follows because anti-transposition is performed by composing the rank preserving operations of transposition and row and column permutations. The third statement follows from the second by:

$$\begin{aligned} r_D(i,j) &= \operatorname{rank} D_i^j - \operatorname{rank} D_i^{j-1} - \operatorname{rank} D_{i+1}^j + \operatorname{rank} D_{i+1}^{j-1} \\ &= \operatorname{rank}(D^\perp)_{n-j}^{n-i} - \operatorname{rank}(D^\perp)_{n-j+1}^{n-i} - \operatorname{rank}(D^\perp)_{n-j}^{n-i-1} + \operatorname{rank}(D^\perp)_{n-j+1}^{n-i-1} \\ &= r_{D^\perp}(n-j, n-i). \end{aligned}$$

□

Proposition 7.10 (Persistence of Dual Filtrations)**.** *Let (X,f) and (X^*,g) be dual filtered complexes with compatible ordering $\sigma_0, \sigma_1, \ldots, \sigma_n$. Then*

1. *(σ_i, σ_j) is a persistence pair in the filtered complex (X,f) if and only if (σ_j^*, σ_i^*) is a persistence pair in (X^*,g), and*
2. *σ_i is essential in (X,f) if and only if σ_i^* is essential in (X^*,g).*

Proof. Lemma 7.9 implies that $r_D(i,j) = r_{D^*}(n-j, n-i)$. Therefore,

$$r_D(i,j) = 1 \Leftrightarrow r_{D^*}(n-j, n-i) = 1.$$

By the Pairing Uniqueness Lemma 2.31, the equivalence above implies that (σ_i, σ_j) is a persistence pair whenever the $(n-j)$-th cell of the dual filtration (X^*,g) is paired with the $(n-i)$-th, thus proving Part (1). For Part (2), Lemma 7.9 also tells us that the following two statements are equivalent.

- Both $r_D(i,j) \neq 1$ and $r_D(j,i) \neq 1$ for all j.
- Both $r_{D^*}(n-j, n-i) \neq 1$ and $r_{D^*}(n-i, n-j) \neq 1$ for all $n-j$.

By Corollary 2.32, this means that σ_i is an essential cell in (X,f) if and only if the $(n-i)$-th cell σ_i^* is essential in the dual filtration (X^*,g). □

Corollary 7.11. *Let (X,f) and (X^*,g) be dual filtered complexes. Then*

1. $$[f(\sigma_i), f(\sigma_j)) \in \mathsf{Dgm}_{\mathbf{F}}^k(f) \Leftrightarrow [g(\sigma_j^*), g(\sigma_i^*)) \in \mathsf{Dgm}_{\mathbf{F}}^{d-k-1}(g),$$

   and

2. $$[f(\sigma_i), \infty) \in \mathsf{Dgm}_\infty^k(f) \Leftrightarrow [g(\sigma_i^*), \infty) \in \mathsf{Dgm}_\infty^{d-k}(g).$$

Proof. Note that for a persistence pair (σ_i, σ_j), found for an ordering compatible with the function f, the birth value is $f(\sigma_i)$ and the death value is $f(\sigma_j)$. The result then follows directly from Theorem 7.10. □

Remark 7.12. It is worth noting that there is a dimension shift between essential and non-essential pairs coming from the fact that the birth cell defines the dimension of a homological feature. For finite persistence pairs, the birth cell changes from σ_i (of dimension k) to σ_j^* (of dimension $d-(k+1)$) in the dual, while for an essential cycle, the birth cell in the dual is σ_i^*.

7.2.1 Dual Discrete Morse Filtrations

We describe explicitly the dual relations between V-paths in X and V-paths in X^*, which allows us to construct a dual filtered discrete gradient field on X^* from a given DGVF on X. We would first like to mention [11], in which the author forsees the beginning of some of the results about dualising vector fields in discete Morse theory that we formalize here.

For a cell complex X with dual X^*, we describe the relations between the (persistent) homology of X and X^*, using discrete Morse theory. We extend some known results in Morse theory to the context of discrete Morse theory. The following definition describes how a dual discrete gradient vector field is induced on the dual complex of X.

Definition 7.13. Let X be a d-dimensional cell complex and X^* its dual. Assume there is a discrete gradient vector field $V = \{(\tau_\lambda^{(k)} \lhd \sigma_\lambda^{(k+1)})\}_\lambda$ on X. Then we define the corresponding *dual discrete gradient vector field* V^* on X^* as $\{(\sigma_\lambda^{*(d-k-1)} \lhd \tau_\lambda^{*(d-k)})\}_\lambda$.

Lemma 7.14 (V-path Duality)**.** *The dual discrete gradient vector field is indeed a discrete gradient vector field. Moreover,*

1. *if $(\tau \lhd \sigma)$ is a pair in (X, V) then $(\sigma^* \rhd \tau^*)$ is a pair in (X^*, V^*), and*
2. *V-paths from τ_0 to σ_n correspond bijectively to V-paths from σ_n^* to τ_0^*.*

Proof. By definition of the dual X^*, V^* also defines a discrete gradient vector field, since we invert the face relation in the dual complex $\sigma^* \lhd \tau^*$ if and only if $\tau \lhd \sigma$. Moreover, $\mathrm{Crit}(V) \cong \mathrm{Crit}(V^*)$, i.e., $\tau \in \mathsf{Crit}(V)$ if and only if $\tau^* \in Crit(V^*)$, as the critical cells are the unpaired ones, and there is a bijection between the pairs.

For the second claim, a k-dimensional V-path

$$(\tau_0 \lhd \sigma_0 \rhd \tau_1 \lhd \sigma_1 \rhd \ldots \rhd \tau_n)$$

corresponds to a $(d-k)$-dimensional V-path

$$(\tau_n^* \rhd \sigma_{n-1}^* \lhd \tau_{n-1}^* \rhd \ldots \rhd \sigma_0^* \lhd \tau_0^*),$$

where each $(\tau_i^* \rhd \sigma_i^*)$ is paired in (X^*, V^*) by the previous argument. □

Remark 7.15. If $f : X \longrightarrow \mathbb{R}$ is a discrete Morse function, then

$$\begin{aligned} f^* : X^* &\to \mathbb{R} \\ \sigma^* &\mapsto -f(\sigma) \end{aligned}$$

defines a discrete Morse function on X^* such that σ^* is f^*-critical if and only if σ is f-critical. Indeed, as $f^*(\sigma^*) = -f(\sigma)$, it follows that $f^*(\sigma^*) \leq f^*(\tau^*)$ if and only if $f(\sigma) \geq f(\tau)$, so

$$f(\tau) \geq f(\sigma) \text{ and } \tau \lhd \sigma \qquad \text{if and only if} \qquad f^*(\tau^*) \leq f^*(\sigma^*) \text{ and } \tau^* \rhd \sigma^*.$$

Hence,

$$\begin{aligned} &|\{\, \sigma^* \in X^* \mid \tau^* \rhd \sigma^* \text{ and } f^*(\tau^*) \leq f^*(\sigma^*) \,\}| \\ &= |\{\, \sigma \in X \mid \tau \lhd \sigma \text{ and } f(\tau) \geq f(\sigma) \,\}| \leq 1, \end{aligned}$$

and

$$|\{\, \rho^* \in X \mid \tau^* \lhd \rho^* \text{ and } f^*(\tau^*) \geq f^*(\rho^*) \,\}| \leq 1.$$

So our function $f^* : X^* \to \mathbb{R}$ is indeed a discrete Morse function. Further, since the cardinalities of the sets of cells above always agree, it follows that dual cell of a critical cell is also critical.

If f induces a discrete gradient vector field V, then f^* induces V^*. Hence, the previous lemma is the discrete Morse theory analogue of the smooth case: for $f : M \longrightarrow \mathbb{R}$ a Morse function on a manifold M, if we consider $-f$, then the flow lines are reversed and the critical values are switched.

Our discrete Morse theory results are summarized in the table below.

	cell complex X	**Dual cell complex X^***
Cell	$\sigma^{(k)}$	$\sigma^{*(d-k)}$
	Filtration	
Filtration	$f : X \longrightarrow \mathbb{R}$ $\sigma \mapsto f(\sigma)$	$f^* : X^* \longrightarrow \mathbb{R}$ $\sigma^* \mapsto -f(\sigma)$
Filtered vector field	$V = \{\, (\tau_\lambda^{(k)} \lhd \sigma_\lambda^{(k+1)}) \,\}_\lambda$	$V^* = \{\, (\sigma_\lambda^{*(d-k-1)} \lhd \tau_\lambda^{*(d-k)}) \,\}_\lambda$
V-path	$(\tau_0 \lhd \sigma_0 \rhd \tau_1 \lhd \ldots \rhd \tau_n)$	$(\tau_n^* \rhd \sigma_{n-1}^* \lhd \ldots \rhd \sigma_0^* \lhd \tau_0^*)$
Critical cells	$\alpha_1, \alpha_2, \ldots \alpha_n$	$\alpha_n^*, \alpha_{n-1}^*, \ldots \alpha_1^*$

7.3 Absolute Persistent Cohomology and Relative Persistent Homology of the Dual

Theorem 7.10 can be proven more algebraically, explicitly exhibiting a shifted chain isomorphism between the absolute persistent homology of X and the relative persistent cohomology of X^* and then applying results of [37]. To prove Theorem 7.10 based on this isomorphism we compose the following isomorphisms of persistent modules, perhaps shifted:

$$\text{absolute persistent homology} \xleftrightarrow{\text{Prop. 2.3 in [37]}} \text{absolute persistent cohomology}$$

$$\xleftrightarrow{\text{shifted chain isomorphism, Theorem 7.17}} \text{relative persistent homology of the dual}$$

$$\xleftrightarrow{\text{Prop. 2.4 in [37]}} \text{absolute persistent homology of the dual.}$$

Let X be a regular cell complex of dimension d. Suppose we have a filtration $\{X_i\}_{i\in I}$ on X and a filtered discrete gradient vector field V that is coherent with the filtration. We denote the n critical cells of (X, V) as $\{\alpha_1, ..., \alpha_n\}$, in order of appearance in the filtration.

Let C_k^n be the restriction of the algebraic Morse chain complex of X to the critical cells of dimension k, and denote by $C_k^i \subset C_k^n$, the vector space of k-dimensional critical cells up to filtration step X_i.

The boundary operator $\partial_k^n : C_k^n \longrightarrow C_{k-1}^n$ restricts to the i-th level (i.e., to the cells in C_k^i), to define the boundary operator

$$\partial_k^i : C_k^i \longrightarrow C_{k-1}^i.$$

We denote the filtered Morse chain complex of (X, V) by:

$$(\mathbb{C},\partial) \ = \quad \begin{tikzcd}
 & {} \arrow[d, hook] & {} \arrow[d, hook] & {} \arrow[d, hook] & \\
... \arrow[r] & C^{i-1}_{k+1} \arrow[r, "\partial^{i-1}_{k+1}"] \arrow[d, hook] & C^{i-1}_{k} \arrow[r, "\partial^{i-1}_{k}"] \arrow[d, hook] & C^{i-1}_{k-1} \arrow[r] \arrow[d, hook] & ... \\
... \arrow[r] & C^{i}_{k+1} \arrow[r, "\partial^{i}_{k+1}"] \arrow[d, hook] & C^{i}_{k} \arrow[r, "\partial^{i}_{k}"] \arrow[d, hook] & C^{i}_{k-1} \arrow[r] \arrow[d, hook] & ... \\
... \arrow[r] & C^{i+1}_{k+1} \arrow[r, "\partial^{i+1}_{k+1}"] \arrow[d, hook] & C^{i+1}_{k} \arrow[r, "\partial^{i+1}_{k}"] \arrow[d, hook] & C^{i+1}_{k-1} \arrow[r] \arrow[d, hook] & ... \\
 & {} & {} & {} &
\end{tikzcd}$$

where $k \in \{0, ...d\}$ stands for the dimension of the cells and $i \in \{1, ...n\}$ for the index in the filtration of chain complexes. We denote the i-th row of this diagram by $\mathbb{C}^i$, the chain complex at filtration step i.

We use the notation $\mathbb{D}^i$ for the chain complex of the i-th step in the filtration of (X^*, V^*). We let $D^n_k =< \alpha^*_n, ...\alpha^*_1 | \dim(\alpha^*_j) = k >$. The restriction of D^n_k to the i first cells of the filtration of X^* is then:

$$D^i_k =< \alpha^*_n, ...\alpha^*_{n-i+1} | \dim(\alpha^*_j) = k > .$$

We set $D^0_k = \{0\}$ for any $k = 0, ...d$.

We now need the notion of relative persistent homology, which was introduced in [37]. The relative filtered chain complex of (X^*, V^*) is defined as

$$(\mathbb{D}^n, \mathbb{D}) = \quad \begin{tikzcd}
 & {} \arrow[d, two heads] & {} \arrow[d, two heads] & {} \arrow[d, two heads] & \\
... \arrow[r] & D^n_{k+1}/D^{i-1}_{k+1} \arrow[r, "\delta^{i-1}_{k+1}"] \arrow[d, two heads] & D^n_k/D^{i-1}_k \arrow[r, "\delta^{i-1}_{k}"] \arrow[d, two heads] & D^n_{k-1}/D^{i-1}_{k-1} \arrow[r] \arrow[d, two heads] & ... \\
... \arrow[r] & D^n_{k+1}/D^{i}_{k+1} \arrow[r, "\delta^{i}_{k+1}"] \arrow[d, two heads] & D^n_k/D^{i}_k \arrow[r, "\delta^{i}_{k}"] \arrow[d, two heads] & D^n_{k-1}/D^{i}_{k-1} \arrow[r] \arrow[d, two heads] & ... \\
... \arrow[r] & D^n_{k+1}/D^{i+1}_{k+1} \arrow[r, "\delta^{i+1}_{k+1}"] \arrow[d, two heads] & D^n_k/D^{i+1}_k \arrow[r, "\delta^{i+1}_{k}"] \arrow[d, two heads] & D^n_{k-1}/D^{i+1}_{k-1} \arrow[r] \arrow[d, two heads] & ... \\
 & {} & {} & {} &
\end{tikzcd}$$

where the maps δ^i_k are now the induced boundaries on the quotient spaces. The i-th row of this diagram is denoted by the pair $(\mathbb{D}^n, \mathbb{D}^i)$.

Let $\mathbb{E}^\bullet = \mathsf{Hom}(\mathbb{C}^{-\bullet}, \mathbb{F}_2)$. We now show that $\mathbb{E}^\bullet \cong (\mathbb{D}^n, \mathbb{D}^{n+\bullet})$, where $\mathbb{C}$ is the filtered Morse chain complex of (X, V) and $\mathbb{D}$ that of (X^*, V^*). Intuitively, the isomorphism follows from the following observation. The coboundary map maps every cell σ to the weighted sum of the cofacets of σ that have already appeared in the filtration. The relative boundary map maps every cell σ^* to the sum of facets of σ^* that have not been quotiented out yet. These two collections are dual to each other. The key argument is based on the following lemma.

Lemma 7.16. *For any $i = 1, ...n$ and any dimension $k = 0, ...d$, there exists a linear isomorphism $\varphi_{i,k} : D^n_{d-k}/D^{n-i}_{d-k} \longrightarrow \mathsf{Hom}(C^i_k, \mathbb{F}_2)$ such that the following diagram commutes.*

$$\begin{tikzcd}
C^i_{(k+1)} \arrow[d, "\cong"] & C^i_{(k)} \arrow[d, "\cong"] & \ni \alpha \arrow[d, mapsto] \\
\mathsf{Hom}(C^i_{k+1}, \mathbb{F}_2) & \mathsf{Hom}(C^i_k, \mathbb{F}_2) \arrow[l, "(\partial^i_{k+1})^T"'] & \ni \hat{\alpha} \\
D^n_{d-k-1}/D^{n-i}_{d-k-1} \arrow[u, "\cong", "\varphi_{i,k-1}"'] & D^n_{d-k}/D^{n-i}_{d-k} \arrow[l, "\delta^i_{d-k}"'] \arrow[u, "\cong", "\varphi_{i,k}"'] & \ni \alpha^* \arrow[u, mapsto]
\end{tikzcd}$$

Proof. We first prove that the quotient vector space D^n_{d-k}/D^{n-i}_{d-k} is isomorphic to $\mathsf{Hom}(C^i_k, \mathbb{F}_2)$. Indeed, by definition,

$$D^n_{d-k}/D^{n-i}_{d-k} = \frac{\langle \alpha^*_n, ...\alpha^*_1 | \dim(\alpha^*_j) = d-k \rangle}{\langle \alpha^*_n, ...\alpha^*_{i+1} | \dim(\alpha^*_j) = d-k \rangle}$$

$$= \langle \alpha^*_i, ...\alpha^*_1 | \dim(\alpha^*_j) = d-k \rangle$$

$$\simeq \langle \alpha_1, ...\alpha_i | \dim(\alpha_j) = k \rangle = C^i_k \simeq \mathsf{Hom}(C^i_k, \mathbb{F}_2),$$

via the map $\alpha^*_i \mapsto \alpha_i \mapsto \hat{\alpha}_i$, where $\hat{\alpha}_i$ denotes the corresponding representative of α_i through the identification $C^i_k \cong \mathsf{Hom}(C^i_k, \mathbb{F}_2)$. Denote this composite by $\varphi_{i,k}$. To show that the diagram commutes, we proceed as follows.

Using the result on V-paths duality of Lemma 7.14, and the fact that $[\alpha^* : \sigma^*] = [\sigma : \alpha]$, we show that $\left(\partial^i_{k+1}\right)^T (\varphi_{i,k}(\alpha^*)) = \varphi_{i,k-1}(\delta^i_{d-k}(\alpha^*))$:

$$\begin{aligned}
\varphi_{i,k-1}\big(\delta^i_{d-k}(\alpha^*)\big) &= \varphi_{i,k-1}\big(\sum_{\sigma^*\in D^n_{d-k-1}/D^{n-i}_{d-k-1}} [\alpha^*:\sigma^*]\sigma^*\big)\\
&= \sum_{\hat\sigma\in\mathsf{Hom}(C^i_{k+1},\mathbb{F}_2)} [\alpha^*:\sigma^*]\hat\sigma\\
&= \sum_{\hat\sigma\in\mathsf{Hom}(C^i_{k+1},\mathbb{F}_2)} [\sigma:\hat\alpha]\hat\sigma\\
&= \big(\partial^i_{k+1}\big)^T(\hat\alpha)\\
&= \big(\partial^i_{k+1}\big)^T(\varphi_{i,k}(\alpha^*)).
\end{aligned}$$

□

It follows in particular that

$$\delta^0_{d-k} : D^n/D^0_{d-k} \longrightarrow D^n/D^0_{d-k-1}$$

is the same as

$$(\partial^n_{k+1})^T : \mathsf{Hom}(C^n_k,\mathbb{F}_2) \longrightarrow \mathsf{Hom}(C^n_{k+1},\mathbb{F}_2).$$

This identification leads to the existence of a filtered chain isomorphism between the cochains of X and the relative chains of X^*.

Theorem 7.17 (Duality of the filtered Morse chain complexes)**.** *Let X and X^* be dual cell complexes and V a filtered gradient vector field on X. Let $(\mathbb{C},\partial)$ be the corresponding Morse chain complex of (X,V) and $(\mathbb{D},\delta)$ that of (X^*,V^*). There exists a shifted filtered chain isomorphism*

$$\mathbb{E}^\bullet \cong (\mathbb{D}^n,\mathbb{D}^{n+\bullet}),$$

where $\mathbb{E}^\bullet = \mathsf{Hom}(\mathbb{C}^{-\bullet},\mathbb{F}_2)$. In particular, it induces a natural *isomorphism between the absolute cohomology of $(\mathbb{C},\partial)$ and the relative homology of $(\mathbb{D},\delta)$.*

Proof. Filtering the the diagram of Lemma 7.16 leads to a new commutative diagram that describes the isomorphism $\mathbb{E}^\bullet \cong (\mathbb{D}^n,\mathbb{D}^{n+\bullet})$ explicitly. The following diagram commutes for any $k=0,...d$, any $i=1,...n$ and any $1-i\le p\le n-i$:

$$\begin{tikzcd}
 & \mathsf{Hom}(C^i_{k+1},\mathbb{F}_2) & & \mathsf{Hom}(C^i_k,\mathbb{F}_2) \arrow[ll] \\
\mathsf{Hom}(C^{i+p}_{k+1},\mathbb{F}_2) \arrow[ur, two heads] & & \mathsf{Hom}(C^{i+p}_k,\mathbb{F}_2) \arrow[ll] \arrow[ur, two heads] \arrow[dd, "\cong"', "\varphi_{i+p,k}"] & \\
 & D^n_{d-k-1}/D^{n-i}_{d-k-1} \arrow[uu, "\cong"', "\varphi_{i,k-1}"] & & D^n_{d-k}/D^{n-i}_{d-k} \arrow[ll] \arrow[uu, "\cong"', "\varphi_{i,k}"] \\
D^n_{d-k-1}/D^{n-i-p}_{d-k-1} \arrow[uu, "\cong"', "\varphi_{i+p,k-1}"] \arrow[ur, two heads] & & D^n_{d-k}/D^{n-i-p}_{d-k} \arrow[ll] \arrow[ur, two heads] &
\end{tikzcd}$$

where the surjective maps are induced by the inclusions $C^i \subset C^{i+p}$. The natural isomorphism between the absolute cohomology of $(\mathbb{C}, \partial)$ and the relative homology of $(\mathbb{D}, \partial)$ is obtained by applying the homology functor to this diagram. □

In particular, there is a bijection between the persistent cohomology pairs of critical cells of (X, V) and the relative persistent homology pairs of (X^*, V^*), defined by

$$(\hat{\alpha}^{(k)}, \hat{\beta}^{(k+1)}) \longleftrightarrow (\beta^{*(d-k-1)}, \alpha^{*(d-k)})$$

and, for essential cycles,

$$(\hat{\gamma}^{(k)}, \infty) \longleftrightarrow (-\infty, \gamma^{*(d-k)}).$$

Here, $\hat{\alpha}$ denotes the representative of a cell $\alpha^{(k)} \in C^n_k$.

Suppose a critical cell $\alpha^{(k)}$ of dimension k appears at step i for the first time in the filtered Morse chain complexes of X and creates a k-cycle, which is destroyed by the appearance of a cell $\beta^{(k+1)}$ at step $i+p$, that is, $\alpha^{(k)}$ is the boundary of $\beta^{(k+1)}$. Hence, we have the pair $(\alpha^{(k)}, \beta^{(k+1)})$ in the absolute persistent homology of X.

In cohomology, that means that $\hat{\alpha}^{(k)}$ disappears after step i, so that the cocycle $\hat{\beta}^{(k+1)}$ is no longer a boundary. When $\hat{\beta}^{(k+1)}$ disappears after step $i+p$, this cocycle disappears as well. The pair in the absolute cohomology of X is then $(\hat{\alpha}^{(k)}, \hat{\beta}^{(k+1)})$.

In the relative sequence of the dual chain complex of X^*, $\beta^{*(d-k-1)}$ disappears first after the step $n-i-p$, so that $\alpha^{*(d-k)}$ becomes a $(d-k)$-cycle. This $(d-k)$-cycle dies when $\alpha^{*(d-k)}$ disappears after step $n-i$. Hence, the pair $(\hat{\alpha}^{(k)}, \hat{\beta}^{(k+1)})$ in the absolute cohomology of X corresponds to the pair $(\beta^{*(d-k-1)}, \alpha^{*(d-k)})$ in the relative homology of the dual X^*.

To provide intuition for why the natural isomorphism relates the absolute cochains of X and the relative chains of X^*, we show what happens to a complex whose dual is not well defined.

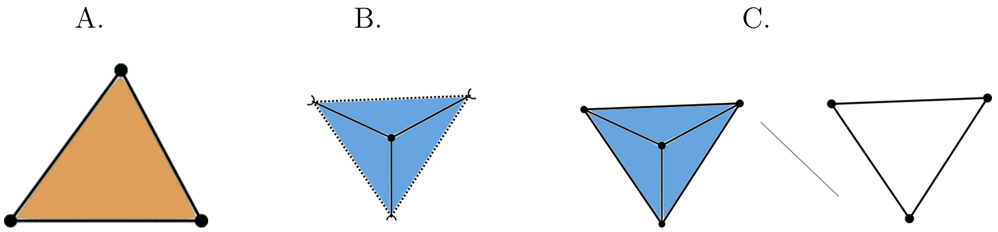


Figure 7.1: A. A non dualizable cell complex made of three 0-cells, three 1-cells and one 2-cell. B. What would be its dual: it does not define a cell complex. C. Another descriptions of the cells in (B).

In Figure 7.1, we show a complex (A) and what its dual should be (B). The middle object (B) is *not* a cell complex – it lacks its boundary. (C) shows another representation of (B) as a set of cells. If we consider the chain complex generated by the cells in (B), we realize that it is the chain complex generated by all the cells of the closure of (B) quotiented by its boundary (C). That illustrates where the relative homology comes from.

Example 7.18. We now illustrate the duality results. We start with a dualizable cell complex X with a function f defined on the vertices and its dual X^* with the function f^* defined on the top-dimensional cells (Figure 7.2). We can extend the values of f to all the cells by assigning a cell the maximum value of its vertices. The dual function f^* is defined on the top-dimensional cells of X^* by: $f^*(\sigma^*) = -f(\sigma)$. To extend it to the full complex X^*, we assign to a cell the minimum of its cofaces. Note that this corresponds to defining f^* by $f^*(\sigma^*) = -f(\sigma)$ on all the cells directly.

An example of a filtered vector field V compatible with the filtration f and the corresponding dual vector field V^* compatible with f^* can be seen in Figure 7.3 and Figure 7.4 respectively. Both of their sets of critical cells are illustrated in Figure 7.5 and Figure 7.6.

We then describe the absolute filtered cochain complex of X and the relative filtered chain complex of X^* in parallel to make the isomorphism of Theorem 7.17 explicit. Finally, we compute the barcodes of the absolute persistent homology of both, providing an example of Theorem 7.11.

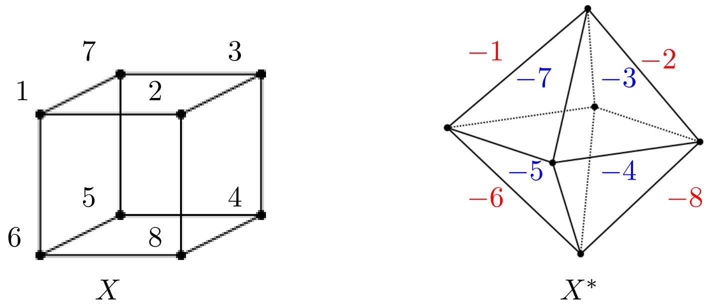


Figure 7.2: On the left, a cellular decomposition of a sphere X with a function f defined on the vertices. We do not represent the 2-cells for visibility reasons (they are the faces of the cube). On the right, the dual complex X^*. We only represent the values of f^* on the top-dimensional cells (faces), to avoid confusion. In red, we show the values of the 2-cells that are in the front, and in blue the 2-cells in the back.

Figure 7.3 shows the filtration on X of Figure 7.2, along with a corresponding filtered vector field V. The corresponding dual filtration on X^*, along with the dual filtered vector field V^*, is shown in Figure 7.4.

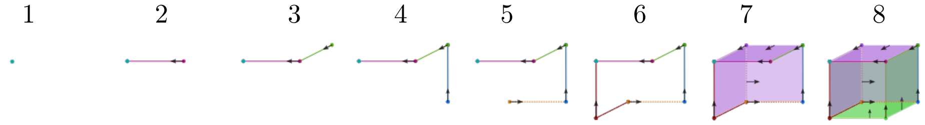


Figure 7.3: A filtration of a cellular decomposition of a sphere X and a filtered vector field V.

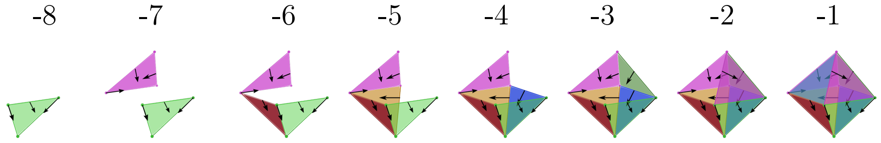


Figure 7.4: The dual filtration of the complex X^* and the dual filtered vector field V^*.

The critical cells of V and V^* are in bijection and they appear in reversed order. Figure 7.5 and 7.6 show the critical cells of X and X^*, that is, the cells that are not paired by the vector fields.

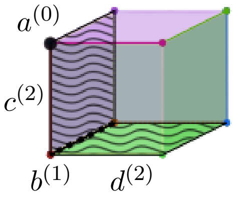


Figure 7.5: A filtered vector field on X respecting the filtration of Figure 7.3 and the corresponding critical cells (the highlighted top left vertex ($a^{(0)}$) and bottom left edge ($b^{(1)}$) and two faces (left hand side: $c^{(2)}$ and bottom: $d^{(2)}$).

The critical cells of V are (in order of appearance):

$$\mathsf{Crit}(V) = \{\, a^{(0)}, b^{(1)}, c^{(2)}, d^{(2)} \,\}.$$

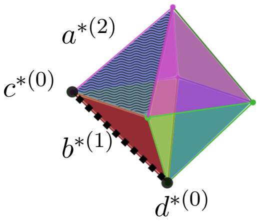


Figure 7.6: The dual complex of Figure 7.5, with the dual critical cells induced by the dual vector field V^* ($a^{*(2)}$ the top 2-cell, $b^{*(1)}$ the bottom edge, $c^{*(0)}$ the left vertex and $d^{*(0)}$ the bottom vertex).

The critical cells of V^* are (in order of appearance):

$$\text{Crit}(V^*) = \{\, d^{*(0)}, c^{*(0)}, b^{*(1)}, a^{*(2)} \,\}.$$

We now build the Morse filtered chain complexes of X and X^*. To illustrate the filtered chain isomorphism between the absolute cochains of (X, f) and the relative chains of (X^*, f^*), we build the respective filtered complexes.

The cochains of the corresponding Morse filtered chain complex of X

$$\begin{tikzcd}
0 & 0 \arrow[l] & 0 \arrow[l] \\
0 \arrow[u, two heads] & 0 \arrow[l] \arrow[u, two heads] & \mathbb{F}_2[\hat{a}] \arrow[l] \arrow[u, two heads] \\
0 \arrow[u, two heads] & \mathbb{F}_2[\hat{b}] \arrow[l] \arrow[u, two heads] & \mathbb{F}_2[\hat{a}] \arrow[l, "\begin{pmatrix}2\end{pmatrix}"'] \arrow[u, two heads] \\
\mathbb{F}_2[\hat{c}] \arrow[u, two heads] & \mathbb{F}_2[\hat{b}] \arrow[l, "\begin{pmatrix}1\end{pmatrix}"'] \arrow[u, two heads] & \mathbb{F}_2[\hat{a}] \arrow[l, "\begin{pmatrix}2\end{pmatrix}"'] \arrow[u, two heads] \\
\mathbb{F}_2[\hat{c}, \hat{d}] \arrow[u, two heads] & \mathbb{F}_2[\hat{b}] \arrow[l, "\begin{pmatrix}1\\1\end{pmatrix}"'] \arrow[u, two heads] & \mathbb{F}_2[\hat{a}] \arrow[l, "\begin{pmatrix}2\end{pmatrix}"'] \arrow[u, two heads]
\end{tikzcd}$$

Applying the homology functor to the previous chain complex, one gets the absolute persistent cohomology module of X:

$$\begin{tikzcd}
0 & 0 & 0 \\
0 \arrow[u] & 0 \arrow[u] & \mathbb{F}_2[\hat{a}] \arrow[u] \\
0 \arrow[u] & \mathbb{F}_2[\hat{b}] \arrow[u] & \mathbb{F}_2[\hat{a}] \arrow[u] \\
0 \arrow[u] & 0 \arrow[u] & \mathbb{F}_2[\hat{a}] \arrow[u] \\
\mathbb{F}_2[\hat{c} + \hat{d}] \arrow[u] & 0 \arrow[u] & \mathbb{F}_2[\hat{a}] \arrow[u]
\end{tikzcd}$$

The corresponding persistence pairs are:

$$(a^{(0)}, \infty), (b^{(1)}, c^{(2)}), (d^{(2)}, \infty).$$

And the absolute persistent cohomology barcode with the f-values:

$$[1, \infty)_0, [6, 7)_1, [8, \infty)_2.$$

The relative chains of the corresponding Morse filtered chain complex of X^*

$$\begin{tikzcd}
\mathbb{F}_2[a^*] \arrow[r, "\begin{pmatrix}2\end{pmatrix}"] \arrow[d, two heads] & \mathbb{F}_2[b^*] \arrow[r, "\begin{pmatrix}1\\1\end{pmatrix}"] \arrow[d, two heads] & \mathbb{F}_2[c^*, d^*] \arrow[d, two heads] \\
\mathbb{F}_2[a^*] \arrow[r, "\begin{pmatrix}2\end{pmatrix}"] \arrow[d, two heads] & \mathbb{F}_2[b^*] \arrow[r, "\begin{pmatrix}1\end{pmatrix}"] \arrow[d, two heads] & \mathbb{F}_2[c^*] \arrow[d, two heads] \\
\mathbb{F}_2[a^*] \arrow[r, "\begin{pmatrix}2\end{pmatrix}"] \arrow[d, two heads] & \mathbb{F}_2[b^*] \arrow[r] \arrow[d, two heads] & 0 \arrow[d, two heads] \\
\mathbb{F}_2[a^*] \arrow[r] \arrow[d, two heads] & 0 \arrow[r] \arrow[d, two heads] & {} \arrow[d, two heads] \\
0 \arrow[r] & 0 \arrow[r] & 0
\end{tikzcd}$$

Applying the homology functor to the previous chain complex, one gets the relative persistent homology module of X^*:

$$\begin{tikzcd}
\mathbb{F}_2[a^*] \arrow[d] & 0 \arrow[d] & \mathbb{F}_2[c^* + d^*] \arrow[d] \\
\mathbb{F}_2[a^*] \arrow[d] & 0 \arrow[d] & 0 \arrow[d] \\
\mathbb{F}_2[a^*] \arrow[d] & \mathbb{F}_2[b^*] \arrow[d] & 0 \arrow[d] \\
\mathbb{F}_2[a^*] \arrow[d] & 0 \arrow[d] & 0 \arrow[d] \\
0 & 0 & 0
\end{tikzcd}$$

The corresponding persistence pairs are:

$$(-\infty, d^{*(0)}), (c^{*(0)}, b^{*(1)}), (-\infty, a^{*(2)}).$$

And the relative persistent homology barcode with the f^*-values:

$$[-\infty, -8)_0, [-7, -6)_1, [-\infty, -1)_2.$$

Applying the bijections of [37], we obtain the absolute persistent homology barcodes of (X, f) and (X^*, f^*):

$$\begin{aligned}\mathsf{Dgm}(X, f) &= \{[1, \infty)_0, [6, 7)_1, [8, \infty)_2\}\\ \mathsf{Dgm}(X^*, f^*) &= \{[-8, \infty)_0, [-7, -6)_0, [-1, \infty)_2\},\end{aligned}$$

illustrating the results of Theorem 7.10.

Chapter 8

Discussion

8.1 Conclusion

This thesis focuses on the inverse problem of reconstructing trees from barcodes. We started by delineating the different spaces of trees and barcodes and described how the main difference between the space of merge trees and phylogenetic trees is the labeling of the leaves.

We discussed two inverse problems. The “real” inverse problem consists of how many trees realize the same barcode. This tree-realization number is computed purely from combinatorial properties of barcodes, in particular their associated permutations. We showed how the TRN can be used to do statistics on barcodes and allows to distinguish biological barcodes from artificial ones. The other inverse problem relates to a biological problem: how to construct artificial neuronal trees that have properties similar to those of neurons. The TNS algorithm, developed in [67], is proven to be stable with respect to the modified bottleneck distance.

Applying the TNS to real neurons’ barcodes leads to trees that mimic neurons. However, to build such an artificial tree, one needs to start from a biological barcode. To be able to study distributions of barcodes, in the hope of one day being able to build artificial barcodes that have identical properties to biological ones, we developed tools to analyse the space of barcodes from a more geometric point of view. The study done in Chapter 4 opens the door to a new way of doing statistics on barcodes. The stratification that one obtains from the Coxeter coordinates looks very similar to combinatorial and geometric objects that one observes in tree spaces (see Section 8.2 below).

The grounds and aims of this thesis were the understanding of the relationship between (merge) trees and barcodes, which has been mainly successfully accomplished through this work. Nonetheless, there is still a lot to learn in this field, the ultimate objective being to generate trees that follow a given set of (biological) properties. We hope that, with the work done in this thesis, we are one step closer to understanding biological barcodes and generating artificial trees that mimic real life properties.

8.2 Perspectives

This work opens up new questions, relating to inverse problems in general as well as the characterization of spaces of trees and barcodes. More specifically, several open questions and perspectives are elaborated here.

Statistics of Barcodes

We showed in Chapter 5 and Chapter 6 that the permutation σ_B associated to a strict barcode B gives nice combinatorial insight into the number of merge trees that have the same barcode. The TRN is derived directly from the permutation, which can also be used to do statistics on barcodes. In this thesis, we focused on the analysis of neurons and demonstrated that the TRN enables us to distinguish different types of neuron barcodes from artificial barcodes. There are other data structures that can be represented as trees, such as rivers, roots, plants, etc. A deeper statistical study of tree-like data using the statistics developed here is a natural follow-up step to consider.

Furthermore, the coordinates of (Theorem 4.22) extend this permutation to any (possibly non-strict) barcode and return a finer invariant than just the permutation. A future direction would be to study this finer invariant defined by $(\bar{b}, \bar{d}, \|v_b\|, \|v_d\|, \sigma_B)$. It might be well-suited for studying statistical questions: the first four elements already have descriptions as averages and standard deviations. The behaviour of the permutation σ_B could be studied using other tools from permutation statistics, such as the number of inversions or descents.

Generating artificial barcodes

One of the main motivations behind this thesis was to develop tools that could lead to the generation of artificial barcodes that mimic real-life data properties. As shown in Section 6.3.6, we are far from a successful method to generate barcodes that have similar behavior to neuron barcodes. The description of the space of barcodes given in Chapter 4 offers new perspectives on this problem. Indeed, this thesis shows that the inverse problem of trees and barcodes is closely related to the perrmutation type of barcodes. Chapter 4 gives a geometric description of the space of barcodes, stratified by permutations. Studying distributions of barcodes in the coordinates defined in Chapter 4 offers potential new ways of generating artificial barcodes.

The final study of Chapter 6, Figure 6.25, shows that there are other combinatorial properties of barcodes that can be explored. They could also be used to develop new methods of generating artificial barcodes.

Combinatorial tools for barcode space

In Chapter 4, we showed that the space $\mathcal{B}_n$ of barcodes with n bars is stratified over the poset of marked double cosets of parabolic subgroups of Sym_n. A question that arises is how this could be extended to the whole space of barcodes, i.e., to the union $\bigcup_{n\in\mathbb{N}} \mathcal{B}_n$. An approach here would be to use appropriate inclusions $\mathcal{B}_m \hookrightarrow \mathcal{B}_n$ for $m \leq n$. Note that on the group level, there are natural injections $\mathrm{Sym}_m \hookrightarrow \mathrm{Sym}_n$ and also on the level of simplicial complexes, $\Sigma(\mathrm{Sym}_n)$ contains copies of $\Sigma(\mathrm{Sym}_m)$ for $m \leq n$.

In a different direction, the description of $\mathcal{B}_n$ in terms of Coxeter complexes allows the rephrasing of these combinatorial questions in more geometric terms. Using this geometric perspective might give new ways for studying invariants and statistics on barcodes.

It would be interesting to see if the geometric and combinatorial tools developed here can help to understand other inverse problems in TDA as the ones in [31, 75]. Since the merge tree-to-barcode problem is related to the symmetric group, it is also natural to ask whether the stratification that we obtain in Theorem 4.21 can be extended to the space of merge trees with n leaves. We discuss this question in the next section.

A lattice version of the inverse problem

There are many similarities between the tree-to-barcode projection and the covering of the subset lattice by the partition lattice, as we saw in Section 5.2.6.

Theorem 5.26 is still in need of a full geometric description that accounts for actual positions and lengths of bars in a barcode and edges in a merge tree. In Section 4.2.2 a novel coordinatization of barcode space was given based on the relation with the symmetric group. However, a similar picture for merge tree space that uses the connection with the partition lattice is unknown. Additionally, the lattice structure on these “skeletonizations” of barcode and merge tree space has not been fully explored. As noted in [62, 82, 93], Möbius inversion provides another way of summarizing topological changes in a filtration, which suggests that inverse problems, lattice theory, and Möbius inversion may occupy a rich intersection of ideas.

Tree-based Topological Loss

On the more applied side, we are interested to see if the TMD can be differentiated to create a tree-based topological loss function for training deep networks. TDA and persistent homology combined with deep learning is an up and coming field with a lot of potential [74]. While many topology-based loss functions are based either on point clouds or on images, it could be useful to develop one based on tree structures using barcodes. Indeed, many applications involve trees, when the underlying structure of the data is a tree or the merge tree of the data is a good summary. While a topology-based loss was developed in collaboration with the Computer Vision Lab at EPFL in [90] to study road networks and neurons in particular, directly using the tree-structure has not yet been considered.

New distances for barcodes

Finally, the modified bottleneck and Wasserstein distances (Theorem 4.23) seem to behave differently from the usual ones. A deeper study of their properties and their potential extension to the space of barcodes (see Theorem 4.25) is a natural next step to consider.

Bibliography

[1] Peter Abramenko and Kenneth S. Brown. *Buildings*, volume 248 of *Graduate Texts in Mathematics*. Springer, New York, 2008.

[2] Henry Adams, Sofya Chepushtanova, Tegan Emerson, Eric Hanson, Michael Kirby, Francis Motta, Rachel Neville, Chris Peterson, Patrick Shipman, and Lori Ziegelmeier. Persistence images: A stable vector representation of persistent homology, 2016.

[3] Aaron Adcock, Erik Carlsson, and Gunnar Carlsson. The ring of algebraic functions on persistence barcodes. *Homology, Homotopy and Applications*, 18, April 2013.

[4] Mohammad Ahsanullah, Valery B. Nevzorov, and Mohammad Shakil. *An introduction to order statistics*, volume 3 of *Atlantis Studies in Probability and Statistics*. Atlantis Press, Paris, 2013.

[5] Federico Ardila and Caroline J. Klivans. The Bergman complex of a matroid and phylogenetic trees. *Journal of Combinatorial Theory, Series B*, 96(1):38–49, 2006.

[6] Giorgio A. Ascoli, Duncan E. Donohue, and Maryam Halavi. Neuromorpho.org: A central resource for neuronal morphologies. *Journal of Neuroscience*, 27(35):9247–9251, 2007.

[7] C. Aslangul, N. Pottier, P. Chvosta, and D. Saint-James. Directed random walk with spatially correlated random transfer rates. *Physical review. E, Statistical physics, plasmas, fluids, and related interdisciplinary topics*, 47 3:1610–1617, 1993.

[8] Nieves Atienza, Luis M. Escudero, María José Jiménez, and M. Soriano-Trigueros. Persistent entropy: a scale-invariant topological statistic for analyzing cell arrangements. *ArXiv*, 1902.06467, 2019.

[9] Dominique Attali, Marc Glisse, Samuel Hornus, Francis Lazarus, and Dmitriy Morozov. Persistence-sensitive simplication of functions on surfaces in linear time. In *TopoInVis'09*, Salt Lake City, United States, 2009.

[10] N. Balakrishnan, Markos V. Koutras, and Konstadinos G. Politis. *Introduction to probability*. Wiley Series in Probability and Statistics. John Wiley & Sons, Inc., Hoboken, NJ, 2020. Models and applications.

[11] Ulrich Bauer. *Persistence in discrete Morse theory.* PhD thesis, Goettingen University, 2011.

[12] R. L. Belton, B. T. Fasy, R. Mertz, S. Micka, D. L. Millman, D. Salinas, A. Schenfisch, J. Schupbach, and L. Williams. Reconstructing embedded graphs from persistence diagrams. *ArXiv*, 1912.08913, 2020.

[13] Louis J. Billera, Susan P. Holmes, and Karen Vogtmann. Geometry of the space of phylogenetic trees. *Adv. Appl. Math.*, 27(4):733–767, Nov 2001.

[14] Anders Björner. Some combinatorial and algebraic properties of Coxeter complexes and Tits buildings. *Advances in Mathematics*, 52(3):173–212, 1984.

[15] Anders Björner and Francesco Brenti. *Combinatorics of Coxeter groups*, volume 231 of *Graduate Texts in Mathematics.* Springer, New York, 2005.

[16] Bea Bleile, Adélie Garin, Teresa Heiss, Kelly Maggs, and Vanessa Robins. The persistent homology of dual digital image constructions. *ArXiv*, 2102.11397, 2021.

[17] Martin R. Bridson and André Haefliger. *Metric spaces of non-positive curvature*, volume 319 of *Grundlehren der Mathematischen Wissenschaften [Fundamental Principles of Mathematical Sciences].* Springer-Verlag, Berlin, 1999.

[18] Daniel Brown and Megan Owen. Mean and variance of phylogenetic trees. *Systematic biology*, 69, August 2017.

[19] Julian Bruggemann. On merge trees and discrete Morse functions on paths and trees. *ArXiv*, 2007.10272v1, 2021.

[20] Benjamin Brück and Adélie Garin. Stratifying the space of barcodes using Coxeter complexes. *ArXiv*, 2112.10571, 2021.

[21] P. Bubenik. Statistical topological data analysis using persistence landscapes. *Journal of Machine Learning Research*, 16:77–102, Jan 2015.

[22] H. Byrne, H. Harrington, R. Muschel, G. Reinert, B. J. Stolz, and U. Tillmann. Topological methods for characterising spatial networks: A case study in tumour vasculature. *ArXiv*, 1907.08711, 2019.

[23] G. Carlsson. Topology and data. *Bulletin of the American Mathematical Society*, 46:255–308, 2009.

[24] Mathieu Carrière, Steve Oudot, and Maks Ovsjanikov. Stable topological signatures for points on 3D shapes. *Computer Graphics Forum*, 34, 2015.

[25] Michael J Catanzaro, Justin M Curry, Brittany Terese Fasy, Jānis Lazovskis, Greg Malen, Hans Riess, Bei Wang, and Matthew Zabka. Moduli spaces of Morse functions for persistence. *Journal of Applied and Computational Topology*, 4(3):353–385, 2020.

[26] David Cohen-Steiner, Herbert Edelsbrunner, and John Harer. Stability of persistence diagrams. volume 37, pages 263–271, Jan 2005.

[27] David Cohen-Steiner, Herbert Edelsbrunner, and John Harer. Extending persistence using Poincaré and Lefschetz duality. *Foundations of Computational Mathematics*, 9:79–103, Feb 2009.

[28] David Cohen-Steiner, Herbert Edelsbrunner, and Dmitriy Morozov. Vines and vineyards by updating persistence in linear time. page 119–126, 2006.

[29] William Crawley-Boevey. Decomposition of pointwise finite-dimensional persistence modules. *Journal of Algebra and its Applications*, 14(05):1550066, 2015.

[30] Hermann Cuntz, Friedrich Forstner, Juergen Haag, and Alexander Borst. The morphological identity of insect dendrites. *PLOS Computational Biology*, 4(12):1–7, Dec 2008.

[31] Justin Curry. The fiber of the persistence map for functions on the interval. *Journal of Applied and Computational Topology*, 2(3):301–321, 2018.

[32] Justin Curry, Jordan DeSha, 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. *ArXiv*, 2107.11212v2, 2021.

[33] Justin Curry, Haibin Hang, Washington Mio, Tom Needham, and Osman Berat Okutan. Decorated merge trees for persistent topology. *ArXiv*, 2103.15804, 2021.

[34] Justin Curry, Sayan Mukherjee, and Katharine Turner. How many directions determine a shape and other sufficiency results for two topological transforms. *ArXiv*, 1805.09782, 2018.

[35] Jacek Cyranka, Konstantin Mischaikow, and Charles Weibel. Contractibility of a persistence map preimage. *Journal of Applied and Computational Topology*, 4(4):509–523, 2020.

[36] Michael W. Davis. The geometry and topology of Coxeter groups. In *Introduction to modern mathematics*, volume 33 of *Adv. Lect. Math. (ALM)*, pages 129–142. Int. Press, Somerville, MA, 2015.

[37] Vin De Silva, Dmitriy Morozov, and Mikael Vejdemo-Johansson. Dualities in persistent (co)homology. *Inverse Problems*, 27, July 2011.

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

[39] Benjamin Delory, Mao Li, Christopher Topp, and Guillaume Lobet. archidart v3.0: A new data analysis pipeline allowing the topological analysis of plant root systems. *F1000Research*, 7:22, 01 2018.

[40] Barbara Di Fabio and Massimo Ferri. Comparing persistence diagrams through complex vectors. In Vittorio Murino and Enrico Puppo, editors, *Image Analysis and Processing — ICIAP 2015*, pages 294–305, Cham, 2015. Springer International Publishing.

[41] P. H. Doyle and D. A. Moran. A short proof that compact 2-manifolds can be triangulated. *Invent. Math.*, 5:160–162, 1968.

[42] Andreas Dress, Katharina Huber, and Mike Steel. A matroid associated with a phylogenetic tree. *Discrete Mathematics and Theoretical Computer Science. DMTCS*, 16, July 2013.

[43] Paul H. Edelman. The Bruhat order of the symmetric group is lexicographically shellable. *Proceedings of the American Mathematical Society*, 82(3):355–358, 1981.

[44] H. Edelsbrunner, D. Letscher, and A. Zomorodian. Topological persistence and simplification. *Discrete Computational Geometry*, 28:511–533, 2002.

[45] Herbert Edelsbrunner and John Harer. Persistent homology, a survey. *Discrete & Computational Geometry - DCG*, 453, Jan 2008.

[46] Herbert Edelsbrunner and Michael Kerber. Alexander duality for functions: the persistent behavior of land and water and shore. In *Proceedings of the twenty-eighth annual symposium on Computational geometry*, pages 249–258, 2012.

[47] Herbert Edelsbrunner and Katharina Ölsböck. Tri-partitions and bases of an ordered complex. *Discrete & Computational Geometry*, pages 1–17, 2020.

[48] P. Erdös, Richard K. Guy, and J. W. Moon. On refining partitions. *Journal of the London Mathematical Society*, s2-9(4):565–570, 1975.

[49] Joseph Felsenstein. The Number of Evolutionary Trees. *Systematic Biology*, 27(1):27–33, March 1978.

[50] Aasa Feragen, Pechin Lo, Marleen de Bruijne, Mads Nielsen, and Francois Lauze. Toward a theory of statistical tree-shape analysis. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Dec 2011.

[51] Aasa Feragen and Tom Nye. Statistics on stratified spaces. In *Riemannian geometric statistics in medical image analysis*, pages 299–342. Elsevier/Academic Press, London, 2020.

[52] Robin Forman. Morse theory for cell complexes. *Advances in Mathematics*, 134(1):90 – 145, 1998.

[53] Andrew Francis and Peter D Jarvis. Brauer and partition diagram models for phylogenetic trees ans forests. *ArXiv*, 2111.15225, 2021.

[54] P. Frosini and C. Landi. Size functions and morphological transformations. *Acta Applicandae Mathematica*, 49:85–104, 1997.

[55] Francis Sir Galton and H. W. Watson. On the probability of the extinction of families. *The Journal of the Anthropological Institute of Great Britain and Ireland*, 4:399–406.

[56] M. Gameiro, Y. Hiraoka, S. Izumi, M. Kramár, K. Mischaikow, and Vidit Nanda. A topological measurement of protein compressibility. *Japan Journal of Industrial and Applied Mathematics*, 32:1–17, 2015.

[57] Marcio Gameiro, Yasuaki Hiraoka, and Ippei Obayashi. Continuation of point clouds via persistence diagrams. *Physica D: Nonlinear Phenomena*, 334:118–132, 2016.

[58] Adélie Garin, Teresa Heiss, K. A. R. Maggs, Beatrice Bleile, and Vanessa Robins. Duality in persistent homology of images. *ArXiv*, 2005.04597, 2020.

[59] Ellen Gasparovic, Elizabeth Munch, Steve Oudot, Katharine Turner, Bei Wang, and Yusu Wang. Intrinsic interleaving distance for merge trees. *ArXiv*, 1908.00063, 2019.

[60] Robert Ghrist, Rachel Levanger, and Huy Mai. Persistent homology and Euler integral transforms. *Journal of Applied and Computational Topology*, 2(1):55–60, 2018.

[61] Gillian Grindstaff and Megan Owen. Geometric comparison of phylogenetic trees with different leaf sets. *ArXiv*, 1807.04235, 2018.

[62] Aziz Burak Gulen and Alexander McCleary. Diagrams of persistence modules over finite posets. *arXiv preprint arXiv:2201.06650*, 2022.

[63] Axel Hultman. The combinatorics of twisted involutions in Coxeter groups. *Transactions of the American Mathematical Society*, 359(6):2787–2798, 2007.

[64] Emile Jacquard, Vidit Nanda, and Ulrike Tillmann. The space of barcode bases for persistence modules. *ArXiv*, 1807.01217, Nov 2021.

[65] Benjamin Johnson and Nicholas Scoville. Merge trees in discrete Morse theory. *ArXiv*, 2007.10272, July 2020.

[66] Sara Kališnik. Tropical coordinates on the space of persistence barcodes. *Found. Comput. Math.*, 19(1):101–129, Feb 2019.

[67] L. Kanari, H. Dictus, A. Chalimourda, W. Van Geit, B. Coste, J. Shillcock, K. Hess, and H. Markram. Computational synthesis of cortical dendritic morphologies. *BioArXiv*, June 2020.

[68] L. Kanari, P. Dłotko, M. Scolamiero, R. Levi, J. Shillcock, K. Hess, and H. Markram. A topological representation of branching neuronal morphologies. *Neuroinformatics*, 16(1):3–13, Jan 2018.

[69] Lida Kanari, Adélie Garin, and Kathryn Hess. From trees to barcodes and back again: theoretical and statistical perspectives. *Algorithms*, 13, 2020.

[70] St Katherine. Review paper: The shape of phylogenetic treespace. *Systematic Biology*, page syw025, June 2016.

[71] Risi Kondor. *Group theoretical methods in machine learning.* PhD thesis, Columbia University, Jan 2008.

[72] Y. Lee, S. Barthel, P. Dłotko, S. M. Moosavi, K. Hess, and B. Smit. High-throughput screening approach for nanoporous materials genome using topological data analysis: Application to zeolites. *Journal of Chemical Theory and Computation*, 14:4427 – 4437, 2018.

[73] Michael Lesnick. The theory of the interleaving distance on multidimensional persistence modules. *Foundations of Computational Mathematics*, 15(3):613–650, 2015.

[74] Jacob Leygonie, Steve Oudot, and Ulrike Tillmann. A framework for differential calculus on persistence barcodes. *arXiv*, 1910.00960, 2021.

[75] Jacob Leygonie and Ulrike Tillmann. The fiber of persistent homology for simplicial complexes. *ArXiv*, 2104.01372, 2021.

[76] Mao Li, Keith Duncan, Christopher Topp, and Daniel Chitwood. Persistent homology and the branching topologies of plants. *American Journal of Botany*, 104, 03 2017.

[77] Mingzhen Li, Sourabh Palande, Lin Yan, and Bei Wang. Sketching merge trees for scientific data visualization. *ArXiv*, 2101.03196v2, 2021.

[78] Albert T Lundell and Stephen Weingram. *The topology of CW complexes.* Springer Science & Business Media, 2012.

[79] Clément Maria, Steve Oudot, and Elchanan Solomon. Intrinsic topological transforms via the distance kernel embedding. *ArXiv*, 1912.02225, 2019.

[80] H. Markram, E. Muller, S. Ramaswamy, M. W. Reimann, M. Abdellah, C. A. Sanchez, A. Ailamaki, L. Alonso-Nanclares, N. Antille, S. Arsever, G. A. Atenekeng Kahou, T. Berger, A. Bilgili, N. Buncic, A. Chalimourda, G. Chindemi, J-D Courcol, F. Delalondre, V. Delattre, S. Druckmann, R. Dumusc, J. Dynes, S. Eilemann, E. Gal, M. Gevaert, J-P Ghobril, A. Gidon, J. Graham, A. Gupta, V. Haenel, E. Hay, T. Heinis, J. B. Hernando, M. Hines, L. Kanari, D. Keller, J. Kenyon, G. Khazen, Y. Kim, J. G. King, Z. Kisvárday, P. Kumbhar, S. Lasserre, J-V Le Bé, B. Magalhães, A. Merchán-Pérez, J. Meystre, B. R. Morrice, J. Muller, A. Muñoz-Céspedes, S. Muralidhar, K. Muthurasa, D. Nachbaur, T. H. Newton, . Nolte, A. Ovcharenko, J. Palacios, L. Pastor, R. Perin, R. Ranjan, I. Riachi, J. Rodríguez, J. L. Riquelme, C. Rössert, K. Sfyrakis, Y. Shi, J. Shillcock, G. Silberberg, R. Silva, F. Tauheed, M. Telefont, M. Toledo-Rodriguez, T. Tränkler, W. Van Geit, J. Villafranca Díaz, . Walker, Y. Wang, S. M. Zaninetta, J. DeFelipe, S. L. Hill, I. Segev, and F. Schürmann. Reconstruction and simulation of neocortical microcircuitry. *Cell*, 163:456–492, 2015.

[81] A. Martino, A. Rizzi, and F. M. F. Mascioli. Supervised approaches for protein function prediction by topological data analysis. *2018 International Joint Conference on Neural Networks (IJCNN)*, pages 1–8, 2018.

[82] Alexander McCleary and Amit Patel. Edit distance and persistence diagrams over lattices. *arXiv preprint arXiv:2010.07337*, 2020.

[83] Konstantin Mischaikow and Vidit Nanda. Morse theory for filtrations and efficient computation of persistent homology. *Discrete & Computational Geometry*, 50(2):330–353, Sep 2013.

[84] Anthea Monod, Bo Lin, Ruriko Yoshida, and Qiwen Kang. Tropical geometry of phylogenetic tree space: A statistical perspective. *ArXiv*, 1805.12400v7, 2018.

[85] D. Morozov, Kenes Beketayev, and G. Weber. Interleaving distance between merge trees. *ArXiv*, 1908.00063, 2013.

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

[87] G. Muszynski, K. Kashinath, V. Kurlin, Michael F. Wehner, and M. Prabhat. Topological data analysis and machine learning for recognizing atmospheric river patterns in large climate datasets. *Geoscientific Model Development*, 12:613–628, 2019.

[88] Vidit Nanda. *Discrete Morse theory for filtrations.* PhD thesis, Rutgers University-Graduate School-New Brunswick, 2012.

[89] James Nation. Notes on lattice theory. Feb 1998.

[90] Doruk Oner, Adélie Garin, Mateusz Koziński, Kathryn Hess, and Pascal Fua. Persistent homology with improved locality information for more effective delineation. *ArXiv*, 2110.06295, 2021.

[91] Steve Oudot and Elchanan Solomon. Barcode embeddings for metric graphs. *ArXiv*, 1712.03630, 2017.

[92] Steve Oudot and Elchanan Solomon. Inverse problems in topological persistence. In *Topological Data Analysis*, pages 405–433. Springer, 2020.

[93] Amit Patel. Generalized persistence diagrams. *Journal of Applied and Computational Topology*, 1(3):397–419, 2018.

[94] Matteo Pegoraro. About the metric space of merge trees with the edit distance. *ArXiv*, 2111.02738, 2021.

[95] Matteo Pegoraro. A metric for tree-like topological summaries. *ArXiv*, 2108.13108, 2021.

[96] Matteo Pegoraro and Piercesare Secchi. Functional data representation with merge trees. *ArXiv*, 2108.13147, 2021.

[97] Hanchuan Peng, Peng Xie, Lijuan Liu, Xiuli Kuang, Yimin Wang, Lei Qu, Hui Gong, Shengdian Jiang, Anan Li, Zongcai Ruan, Liya Ding, Chao Chen, Mengya Chen, Tanya L. Daigle, Zhangcan Ding, Yanjun Duan, Aaron Feiner, Ping He, Chris Hill, Karla E. Hirokawa, Guodong Hong, Lei Huang, Sara Kebede, Hsien-Chi Kuo, Rachael Larsen, Phil Lesnar, Longfei Li, Qi Li, Xiangning Li, Yaoyao Li, Yuanyuan Li, An Liu, Donghuan Lu, Stephanie Mok, Lydia Ng, Thuc Nghi Nguyen, Qiang Ouyang, Jintao Pan, Elise Shen, Yuanyuan Song, Susan M. Sunkin, Bosiljka Tasic, Matthew B. Veldman, Wayne Wakeman, Wan Wan, Peng Wang, Quanxin Wang, Tao Wang, Yaping Wang, Feng Xiong, Wei Xiong, Wenjie Xu, Zizhen Yao, Min Ye, Lulu Yin, Yang Yu, Jia Yuan, Jing Yuan, Zhixi Yun, Shaoqun Zeng, Shichen Zhang, Sujun Zhao, Zijun Zhao, Zhi Zhou, Z. Josh Huang, Luke Esposito, Michael J. Hawrylycz, Staci A. Sorensen, X. William Yang, Yefeng Zheng, Zhongze Gu, Wei Xie, Christof Koch, Qingming Luo, Julie A. Harris, Yun Wang, and Hongkui Zeng. Brain-wide single neuron reconstruction reveals morphological diversity in molecularly defined striatal, thalamic, cortical and claustral neuron types. *BioArXiv*, 2020.

[98] T. Kyle Petersen. A two-sided analogue of the Coxeter complex. *Electronic Journal of Combinatorics*, 25(4):Paper 4.64, 28, 2018.

[99] M. Pont, J. Vidal, J. Delon, and J. Tierny. Wasserstein distances, geodesics and barycenters of merge trees. *IEEE Transactions on Visualization Computer Graphics*, (01):1–1, Sep 5555.

[100] Alexander Postnikov. Permutohedra, associahedra, and beyond. *Int. Math. Res. Not. IMRN*, (6):1026–1106, 2009.

[101] Vanessa Robins. *Computational Topology for Point Data: Betti Numbers of α-Shapes*. PhD thesis, Colorado State University, 2002.

[102] Primoz Skraba and Katharine Turner. Wasserstein stability for persistence diagrams. *ArXiv*, 2006.16824, 2021.

[103] Elchanan Solomon, Alexander Wagner, and Paul Bendich. From geometry to topology: Inverse theorems for distributed persistence. *ArXiv*, 2101.12288, 2021.

[104] A. Verri, C. Uras, P. Frosini, and M. Ferri. On the use of size functions for shape analysis. *Biological Cybernetics*, 70:99–107, 2004.

[105] Chenguang Xu. A correspondence between Schubert cells and persistence diagrams. *Master thesis, Kyoto university, Supervisor: Yasuaki Hiraoka*, 2020.

[106] Lin Yan, Talha Bin Masood, Farhan Rasheed, Ingrid Hotz, and Bei Wang. Geometry-aware merge tree comparisons for time-varying data with interleaving distances. *ArXiv*, 2107.14373, 2021.

[107] Lin Yan, Yusu Wang, E. Munch, Ellen Gasparovic, and Bei Wang. A structural average of labeled merge trees for uncertainty visualization. *IEEE Transactions on Visualization and Computer Graphics*, 26:832–842, 2020.

[108] A. Zomorodian and G. Carlsson. Computing persistent homology. 2004.