POLITECNICO DI TORINO

DIPARTIMENTO DI SCIENZE MATEMATICHE

Corso di Laurea Magistrale in Ingegneria Matematica

Master's Degree in Mathematical Engineering

Tesi di Laurea Magistrale

Master's Degree Thesis

Topological Data Analysis and Persistent Homology

Supervisor

Prof. Francesco Vaccarino

Candidate

Carla Federica Melia

A.Y. 2017/2018

Graduation Session of December 2018

Acknowledgements

I would like to express my sincere gratitude to my supervisor, Professor Francesco Vaccarino, for his continuous support and motivation, for his patience and deep knowledge. His guidance helped me in the research and writing of this thesis. The responsibility for any inaccuracy in this paper is to be ascribed only to me.

I thank also my family, my friends and my colleagues for supporting me spiritually throughout writing this thesis.

Thanks you all!

Carla Federica

Ringraziamenti

Desidero ringraziare il relatore, il Professore Francesco Vaccarino, per la grandissima disponibilità, per i confronti costruttivi, per gli spunti e per le cose insegnate e condivise che porterò sempre con me. Lo ringrazio anche per avermi introdotto a questo bellissimo argomento e aiutato a comprenderlo più approfonditamente. A me spetta la responsabilità per eventuali imprecisioni contenuti in questo testo.

Ringrazio di cuore mia madre, Gladys, per avermi sostenuta e per il suo importante esempio. Non sarebbe stato possibile raggiungere questo traguardo senza di lei, le dedico quindi questo lavoro.

Ringrazio mio nonno Federico e i miei familiari per aver creduto in me, Morgan per la sua dolcezza e le sue incoraggianti parole e Giorgio per i suoi preziosi consigli e la sua simpatia.

Ognuno di voi mi ha a suo modo aiutato.

Grazie di tutto!

Carla Federica

Astract

Topological Data Analysis (TDA) uses algebraic topology, statistics and computer science techniques to infer robust features of complex datasets eventually corrupted by noise.

This thesis focuses on Persistent Homology (PH) technique and its purposed are:

1. to provide a satisfying explanation of TDA and PH fundamentals, tools and topics,
2. to analyse the robustness and the reliability of the inferred features with the statistical interpretation of the results,
3. to practically implement some TDA techniques on some study cases.

In PH technique, the input is assumed to be a finite set of elements coming with a notion of distance between them. The elements are mapped into a PCD that is completed by building a nested family of simplicial complexes on it.

Homotopy groups are algebraic objects that intuitively measures the amount of "n-dimensional holes" of a space but a more computable alternative are the homology groups whose ranks represents the Betti numbers. The first three of them count respectively the number of connected components, of holes and of voids in a topological space.

With PH we study the homology of a filtered simplicial complex as a single algebraic entity. Its features can be then analysed using its barcode representation and this is formally justified by the Structure Theorem. Then, the most persistent features can be easily detected and separated from topological noise using statistical methods as the Bootstrap.

GUDHI has been highlight as one of the best available open-source libraries for TDA computation. To analyse the topological information of different datasets, a console application was implemented using GUDHI in Python, TDA in R and QlikView.

Contents

1. Introduction

Topological Data Analysis (TDA) is a branch of applied mathematics that uses notions and techniques of a miscellaneous set of scientific fields. Among these, algebraic topology, data analysis, computer science and statistics are included. Its resulting tools allow to infer relevant and robust features of complex datasets that can present rich structures eventually corrupted by noise and incompleteness[74].

The works[62][38] of Edelsbrunner, Letscher, Zomorodian, and Carlsson, published between 2002 and 2005, are considered the first milestones of this field. However, it already supplies mature methods that had been successfully used in data mining.

This thesis focuses on *persistent homology* (PH) technique, but there are also other methods in TDA such as the Euler calculus and cellular sheaves[74]. See paper [68] for some examples. The purposes of this thesis are:

1. To provide a concise but satisfying explanation of TDA and PH fundamentals, tools and topics.
2. To analyse the robustness and the reliability of the inferred features with the statistical interpretation of the results.
3. To implement some TDA techniques using a mixture of Python[110], R[84] and QlikView[107].

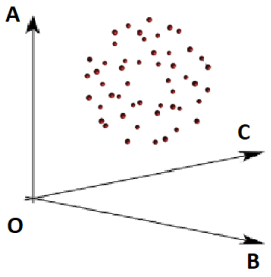
TDA aims to infer properties of the "shape" of data. To get an intuitive idea of what "shape" of data is referred to, see the pipeline in Figure 1.1.

TDA methods can be divided[103] in:

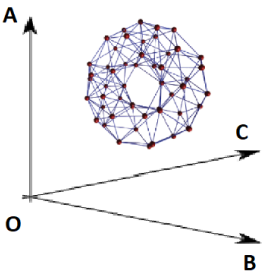
- *Topological Data Compression* to represent the shape of data,
- *Topological Data Completion* to "measure" the shape of data.

A	B	C
506.000000	506.000000	506.000000
68.574901	3.795043	9.549407
28.148861	2.105710	8.707259
2.900000	1.129600	1.000000
45.025000	2.100175	4.000000
77.500000	3.207450	5.000000
94.075000	5.188425	24.000000
100.000000	12.126500	24.000000

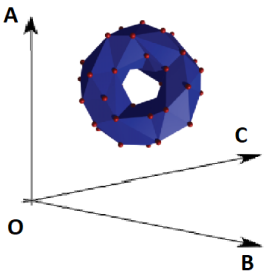
(a)



(b)



(c)



(d)

Figure 1.1: Given a dataset in (a), we can represent it in a 3D Euclidean coordinate system (b). Suppose to get the graph in (c) by connecting with edges nearby points. If we approximate this shape with a continuous one, by adding a face in correspondence of n-uples of nearby points, we can detect a torus-shaped set (d). Torus image from [34].

This thesis focuses on the second family of methods, but now a brief overview of the first one is provided. This is meant help the reader understand some of the limitations of more common methods and how topology can avoid, or at least dampen, them.

1.1 Topological Data Compression

Topological Data Compression algorithms aim at representing a collection of high dimensional clouds of points (PCDs) through graphs. These techniques relies on the fact that an object made of lots or infinitely many points, like the complete circle in Figure 1.2, can be approximated using only some nodes and edges[28].

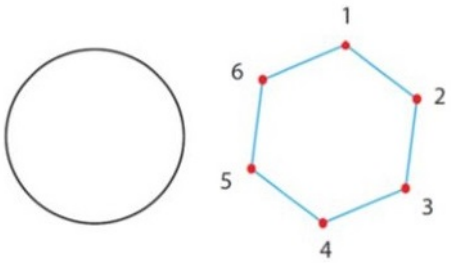


Figure 1.2: Compressed representations idea, image from [31].

Consider for example the "Y"-shape detectable in Figure 1.3.

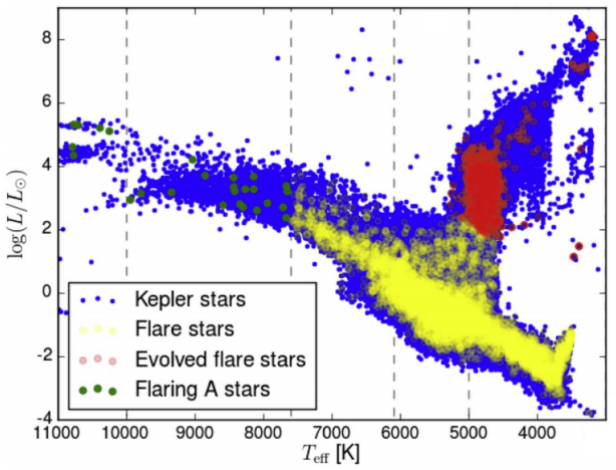


Figure 1.3: HR diagram with the flare stars indicated. The central panel has stellar temperature as the horizontal axis, while the vertical axis shows the luminosity. Image from [58].

This shape occurs frequently in real data sets. It might represent a situation where the core corresponds to the most frequent behaviors, and the tips of the flares to the extreme ones[30]. However, classical models such as linear ones and clustering methods can't detect satisfactorily it. Another not trivial but important shape is the circular one. See the example in Figure 1.6 (right).

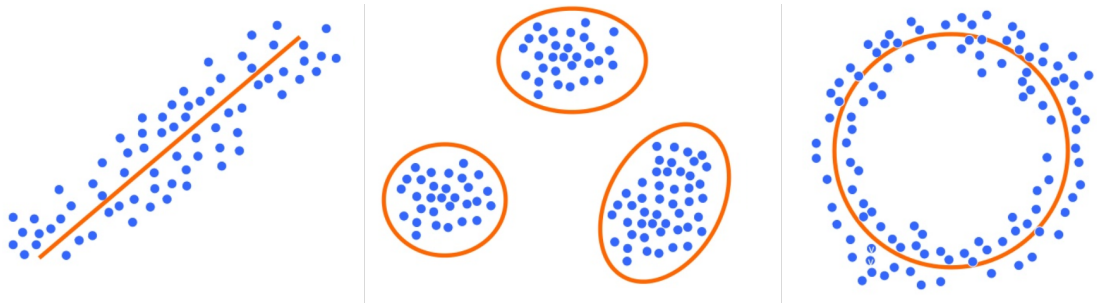


Figure 1.4: Clouds of point modelizable by a linear model (left), clustering (middle) and a circular model (right) respectively. Images from [29].

Loops can denote periodic behaviors. For example, in the Predatory-Prey model in Figure 1.5, the circular shapes are caused by the cyclicity of the described biological system.

In a dynamic system, an *attractor* is a set towards which it tends to evolve after a sufficiently long time. System values that get close enough to the attractor ones have to remain close to them, even if slightly disturbed. A trajectory of a dynamic system on an attractor must not satisfy any particular property, but it's not unusual for it to be periodic and to present loops. See in Figure 1.5.

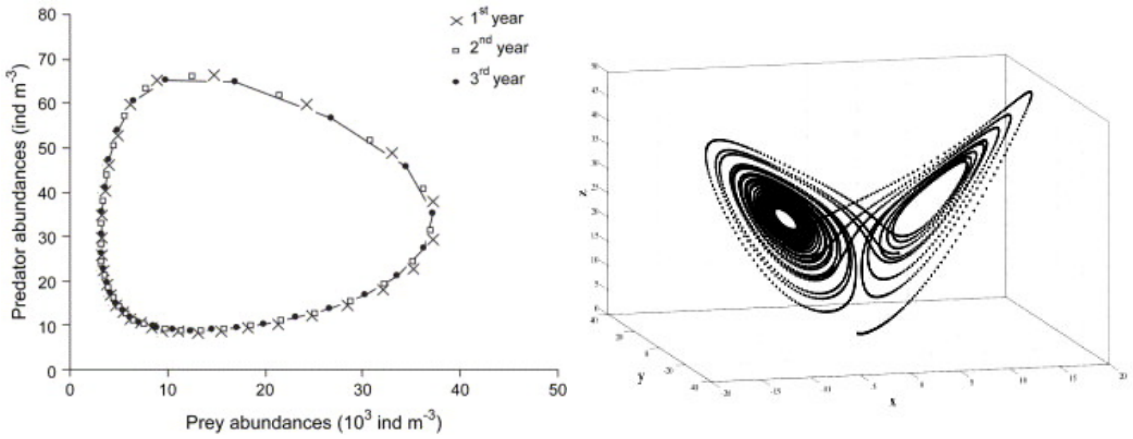


Figure 1.5: On the left, the Predatory-Prey model for some specific choice of parameters. Image from [41]. On the right, the attractor of Lorenz system. Image rearranged from [98].

Consider, moreover, the great amount of not trivial geometric information carried by biomolecules and their importance for the analysis of their stability[63].

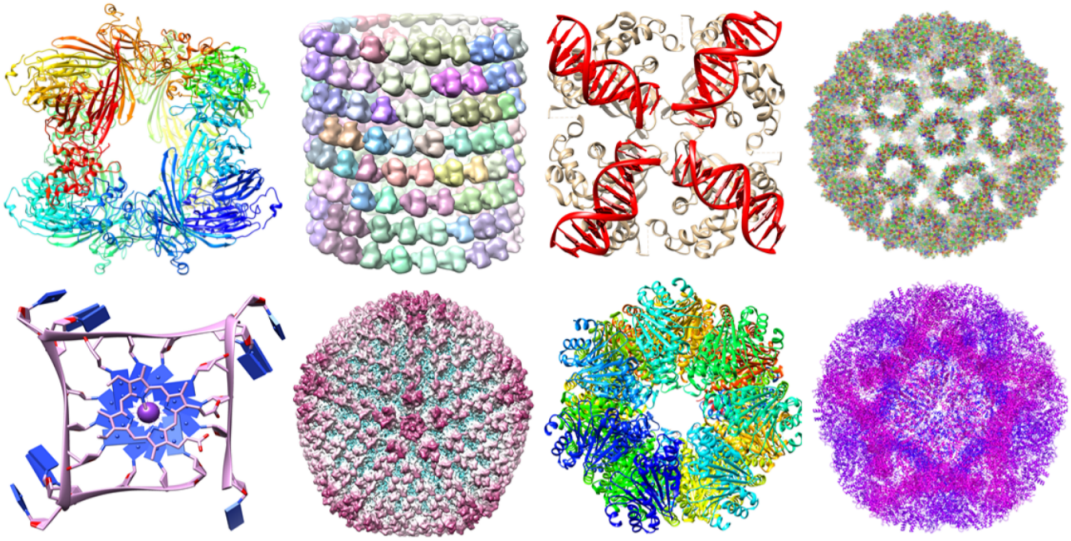


Figure 1.6: Examples of biomolecular systems. Image from [64].

Topological Data Compression can detect these shapes easily and the main algorithm in this area is *Mapper*. Its steps can be described as follows[88]:

1. A PCD representing a shape is given.
2. It is covered with overlapping intervals by coloring the shape by filter values.
3. It is broken into overlapping bins.
4. The points in each bin are collapsed into clusters. Then a network is built representing each cluster by a vertex and drawing an edge when clusters intersect.

See Figure 1.7.

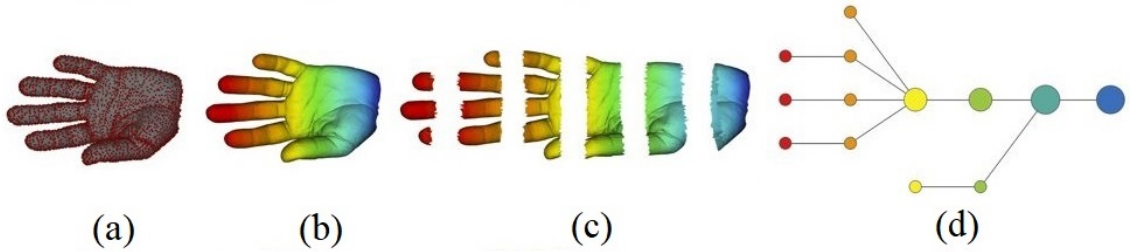


Figure 1.7: (a) A 3D object represented as a PCD. (b) A filter value is applied and the object is colored by the values of the filter. (c) The dataset is binned into overlapping groups. (d) Each bin is clustered and a network is built. Image from [3].

With topological data compression we can achieve a compressed representation of all trivial and not trivial data shapes going beyond the results proposed by predetermined structure models such as linear models or clustering.

The circle on the right of Figure 1.6 could be mapped by the Mapper as shown in Figure 1.8.

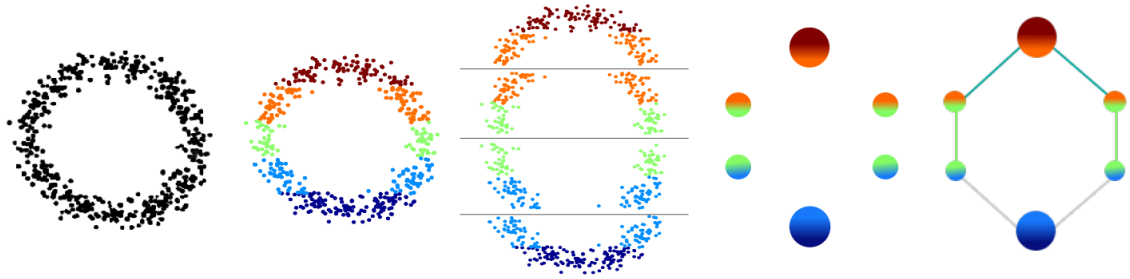


Figure 1.8: The Mapper approach is applied to a circular PCD. Image from [10].

Some achievements of the Mapper are now reported.

1.1.1 New Subtype of Cancer Discovered

In work published in 2011 by Nicolau, Carlsson, and Levine[37], a new subtype of breast cancer was discovered[95] using the Progression Analysis of Disease (PAD), an application of the Mapper that provided an clear representation of the dataset.

The used dataset describes the gene expression profiles of 295 breast cancer tumors[5]. It has 24.479 attributes and each of them specifies the level of expression of one gene in a tissue sample of the corresponding tumor.

The researchers discovered the three-tendril "Y"-shaped structure shown in Figure 1.9.

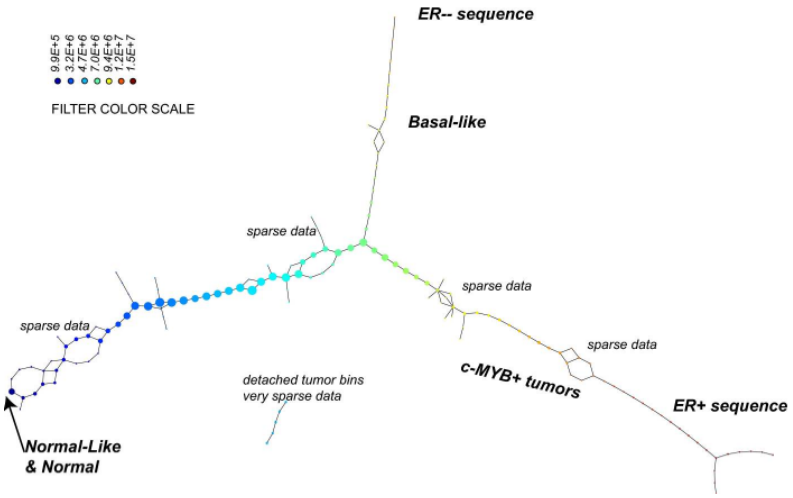


Figure 1.9: The topological network for the dataset of gene expression profiles of breast cancer patients. Image from [95].

In addition, they found that one of these tendrils decomposes further into three clusters. One of these three clusters corresponds to a distinct new subtype of breast cancer that they named c-MYB+.

A standard approach to the classification of breast cancers, based on clustering, divides breast cancers into five groups and these results suggested a different taxonomy not accounted before. In particular, the dendrogram of this dataset is shown in Figure 1.10.

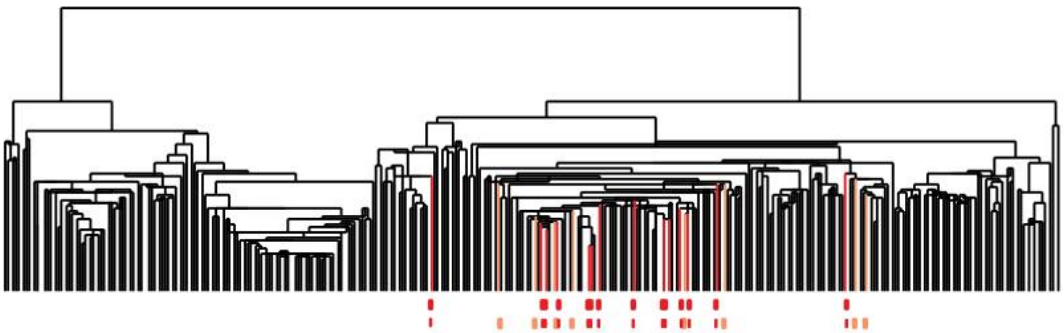


Figure 1.10: Dendrogram of the cancer dataset. The bins defining the c-MYB+ group are marked in red. Image from [95].

The c-MYB+ tumors are scattered among different clusters and there are many non-members of their group which lie in the same high-level cluster. Although, PAD was able to extract this group that turns out to be both statistically and clinically coherent.

The problem is that clustering breaks data sets into pieces, so it can break things that belong together apart.

1.1.2 A New Model Validation Technique

In Figure 1.11, an example of model validation using TDA is shown.

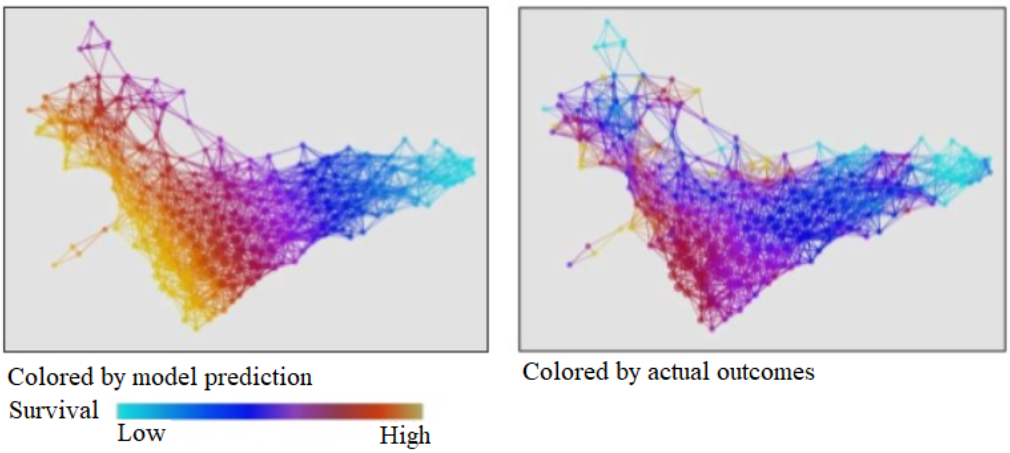


Figure 1.11: Dataset about information of some cancer patients represented using the Mapper. On the left, the color is based on a score that indicated how much it is possible for a patient to die. On the right, by the actual state of the patient. Image rearranged from [29].

We can see that the status of the terminal patients in the right part of the graph has been adequately predicted. This has not happened in regards to the patients in the top left of the graph: their actual status is indeed terminal.

Examining the data, it was found out that those patients had not answered some questions about energy and movements in the questionnaire used to build the model[9]. Thanks to this immediate graphical representation, it was easy to understand that, to improve the model, the patients who did not fill that specific part of the questionnaire had to be studied separately.

1.1.3 Improved Machine Learning Algorithms

Ayasdi[7] is a machine intelligence software company that offers to organizations solutions able to analyze data and to build predictive models from them[9].

In particular, the Ayasdi system runs many different unsupervised and supervised machine learning algorithms on data, finds and ranks best fits automatically and then applies TDA to find similar groups in the results. See Figure 1.12.

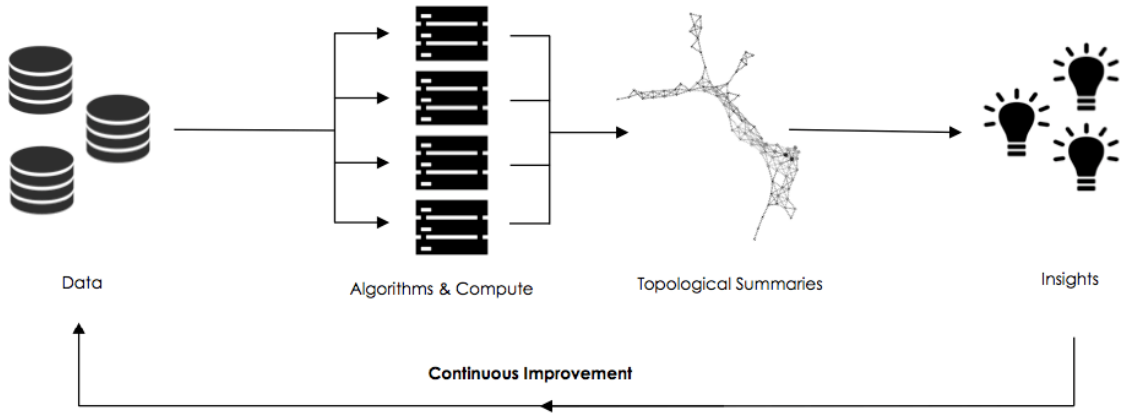


Figure 1.12: Ayasdi approach. Image from [113].

This is performed to reduce the possibility of missing critical insights by reducing the dependency on machine learning experts choosing the right algorithms[10].

This methodology has lead to many achievements, for example:

- DARPA (Defense Advanced Research Projects Agency) used Ayasdi Core to analyze acoustic data tracks. The analysis identified signals that had been previously classified from traditional signal processing methods as unstructured noise[9].
- Using TDA on the portfolio of a G-SIB institution, the bank was able to identify performance improvements by 103bp in less than two weeks. This was worth over 34 million annually, despite this portfolio had been heavily analyzed previously[26].
- TDA was used on the care process model for pneumonia of the Flagler Hospital. Ayasdi methodology could extract nine potential pneumonia care pathways, each with distinctive elements. This represented a potential savings of more than $400K while delivering better care[6].

An example of output achieved with Ayasdi softwares is shown in Figure 1.13.

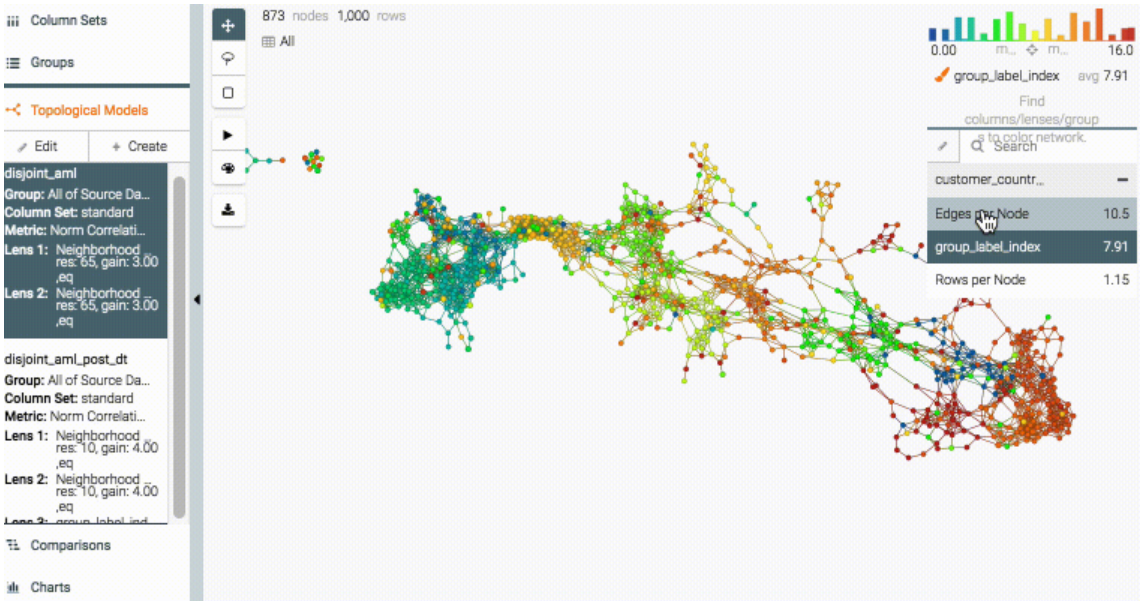


Figure 1.13: Ayasdi anti-money laundering application example preview. Image from [8].

1.1.4 Mapper Open Points

The Mapper algorithm is simple but various choices that are left to the user[48]:

- *The filter function*: sometimes the centrality and the eccentricity functions appears to be good choices that do not require any specific insight about the data.

- *The covering*: when the filter function is a real-valued function, the cover can be chosen to be a set of regularly spaced intervals of equal length $r > 0$. A classical strategy to choose r consists in exploring a range of parameters and chose the ones that turn out to provide the most informative output.

- *The clusters*: a common strategy consists in applying a clustering algorithm, chosen by the user, on each bin. A second strategy consists in building a neighboring graph on top of the data and, for each bin, considering the connected components of the corresponding subgraph.

These open points can pave the way for an interesting discussion, but in this thesis we will focus on the topological summaries provided by Topological Data Completion methods that use persistent homology.

1.2 Topological Data Completion

Topological data completion aims at detecting and counting properties of the shape of data that are preserved under continuous deformations such as crumpling, stretching, bending and twisting, but not gluing or tearing. Components, loops and voids are examples of these invariant properties.

For example, an "A"-shape has one loop and a "B"-shape has two. This property does not change despite deforming continuously the letters, see Figure 1.14.

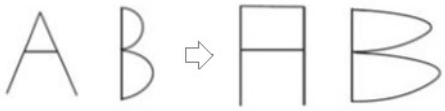


Figure 1.14: Invariance under deformation idea, image from [31].

PCDs, digital images, level sets of real-valued functions and networks can be studied with topological data completion[74] and it generally consists of the following steps, also shown in Figure 1.15.

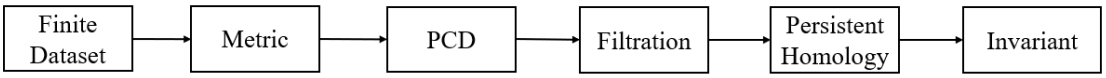


Figure 1.15: Pipeline of TDA.

1. The input is assumed to be a finite set of elements coming with a notion of distance between them. The choice of the metric is critical to revealing the topological features of the data.

2. The elements are mapped into a PCD.

3. The PCD is completed by building "continuous" shape on it called *complex*. For robustness reasons, a nested family of complexes, called *filtration*, is generally built. This is often a *simplicial* filtration.

4. *Homology* associates to complexes some algebraic groups that allow to count beyond individual occurrences: they will be used to count equivalence classes of occurrences[28]. Homology is used because it is based on a well-understood theoretical framework, is computable via linear algebra and is robust to small perturbations[74].

5. Finally, the most persistent features are detected using PH. They are supposed to represent true characteristics of the underlying space rather than noise[31].

In Figure 1.16 proposed in [68], this pipeline is summarized.

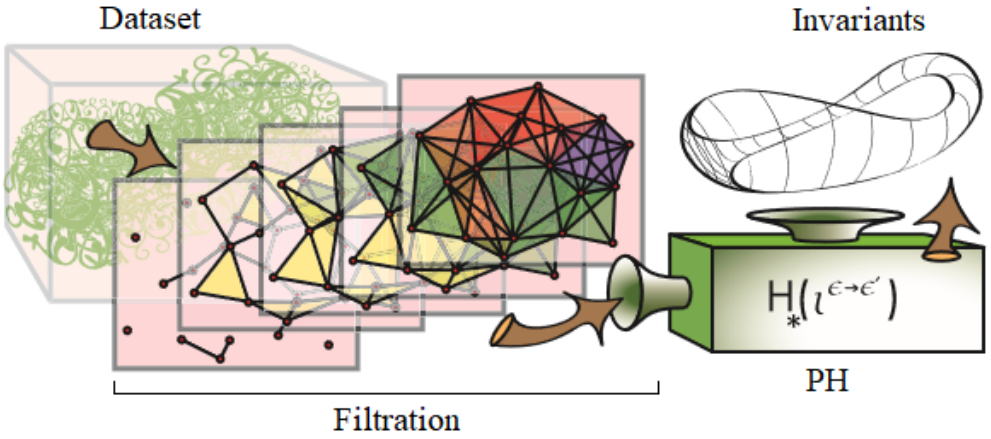


Figure 1.16: PH of a simplicial approximation finds hidden structures in large data sets. Image rearranged from [68].

Using topology, TDA represents data in such a way there is no dependence on a coordinate system, while preserving the metric information[28]. Suppose we are interested in identifying the loop shown in Figure 1.17. If the coordinates are stretched out, with TDA our ability to detect the loop won't be affected. This allows, for example, to analyse data originated from different technologies.

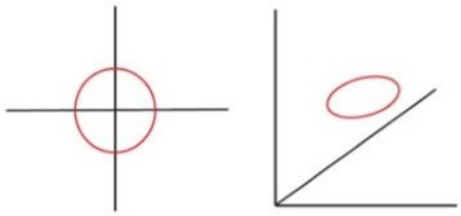


Figure 1.17: Coordinate freeness idea, image from [31].

With topological data completion we can automatically detect and count components, loops and voids benefiting of coordinate and deformation invariance.

A very incomplete list of successful applications of PH includes viral evolution[32], bacteria classification[104], propagation on networks[117], analysis of disease progression[37], complex network[83], sensor networks[72], cosmic web[57], signal analysis[79], image analysis[43], shape study[92], material analysis[56] and fractal

geometry[97]. Some examples are now given and new applications appear progressively frequently[74].

1.2.1 Improved CT Reconstruction for Computed Tomography

To reduce the risk of radiation to patients, compressed sensing computed tomography using sparse projection views has been extensively investigated. However, an analytic reconstruction approach results in severe streaking artifacts due to the low number of projection views. Moreover, CS-based iterative approach is computationally expensive.

In 2016, to address these issues, the KAIST[78] developed a deep residual learning approach for sparse-view reconstruction based on a PH. The proposed approach provided significantly better image reconstruction performance (see Figure 1.18) with orders of magnitude faster computational speed over the image learning.

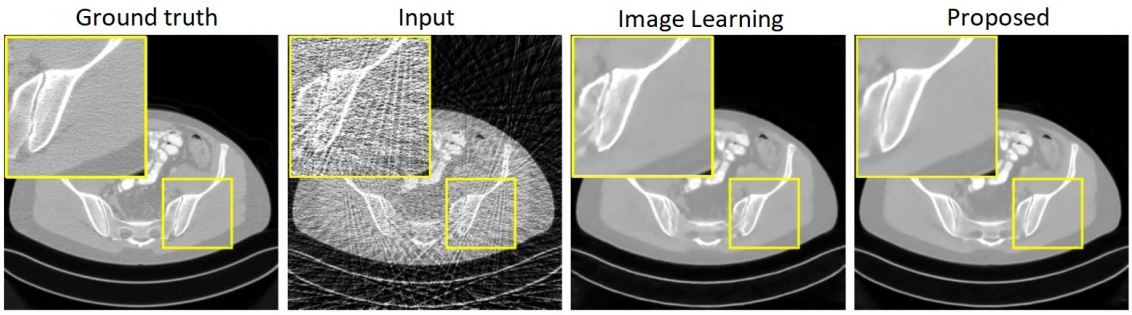


Figure 1.18: Comparison results of TDA-improved learning method and "classical" image learning from 64 view reconstruction input data. Image from [78].

1.2.2 More Effective Brain Networks Analysis

Traditionally, the structure of very complex networks has been studied through their statistical properties and metrics. However, the interpretation of functional networks can be hard. This had motivated the widespread of thresholding methods that risk overlooking the weak links importance. In order to overcome these limits, in the paper *Homological scaffolds of brain functional networks*[24], an efficient alternative analysis of brain functional networks was provided.

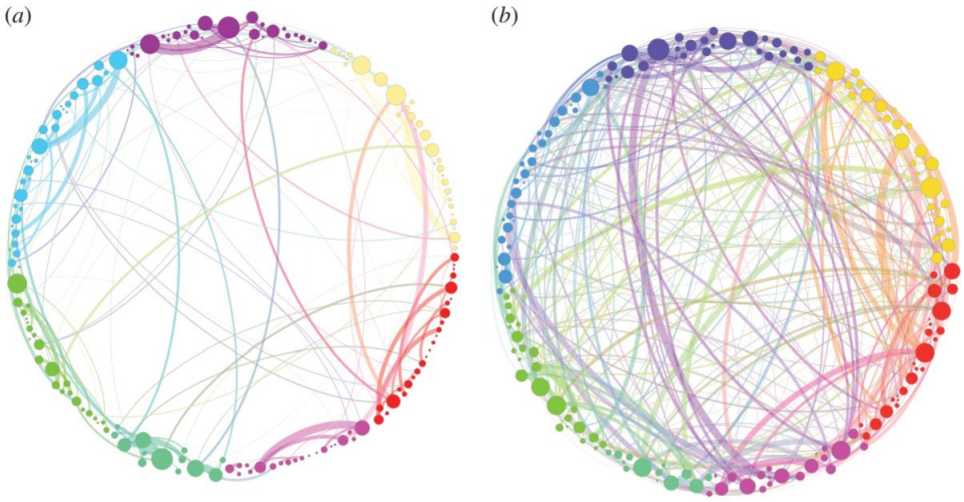


Figure 1.19: Simplified visualization of the persistence homological scaffolds. Only the links heavier than 80 are shown. Colors represent communities obtained by modularity optimization. In (a) the placebo baseline is shown, in (b) the post-psilocybin structure one. The links widths are proportional to their weight and the diameter of the nodes to their strength. Image from [24].

The detected topological information was leveraged to define the *homological scaffolds*, objects designed, on one hand, to represent compactly the homological features of the correlation network and, on the other, to allow the study of their homological properties with networks methods.

These tools were applied to compare functional brain activity after intravenous infusion of placebo and psilocybin, a psychoactive component. The results, consistently with psychedelic state medical descriptions, show that the post-psilocybin homological brain structure is characterized by many transient structures of low stability and of a low number of persistent ones. This means that the psychedelic state is associated with less constrained and more inter-communicative brain activities[24]. See Figure 1.19.

In [50], to overcome the threshold problem, TDA was used to model all brain networks generated over every possible threshold. The evolutionary changes in the number of connected components are displayed in Figure 1.20.

In [36], PH was successfully used to find hidden structures in experimental data associated with the V1 visual cortex of certain primates.

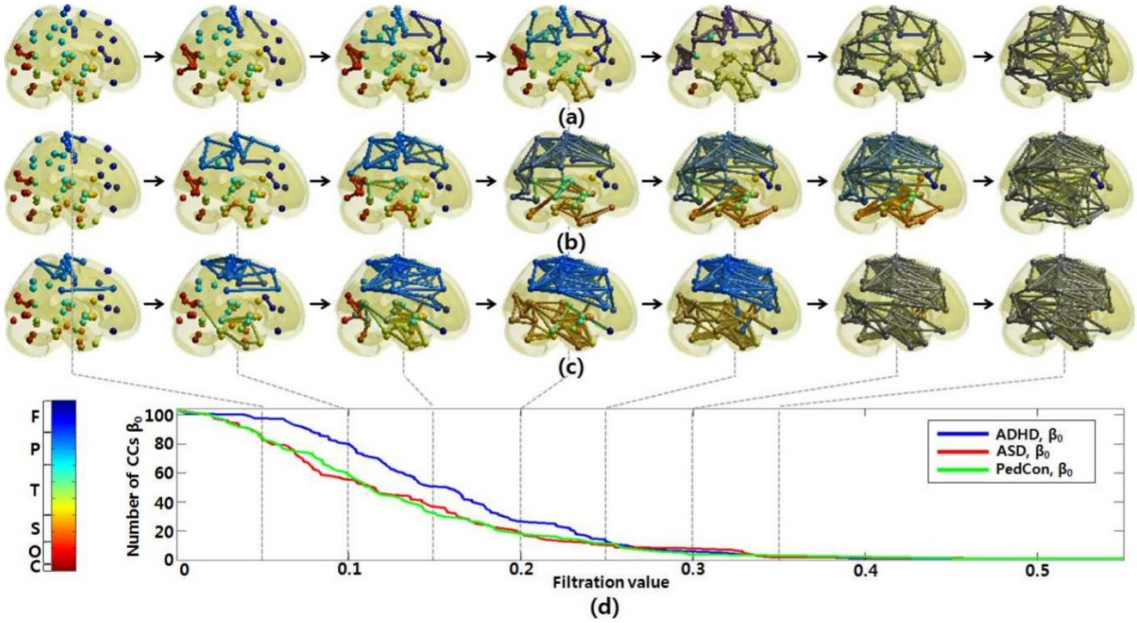


Figure 1.20: Graph filtration of (a) ADHD, (b) ASD and (c) PedCon at the filtration values $\varepsilon = 0.1, 0.15, 0.2, ..., 0.45$. *The color of nodes at* $\varepsilon = 0$ *is shown in the colorbar. If the nodes belong to the same connected component, they are colored identically. The number of connected components is displayed in the graph (d). Image from [50].*

1.2.3 Combined Characterization of both Vertical and Horizontal Evolution of Viral Genomes

Evolution is mediated not only by random mutations over a number of generations (vertical evolution), but also through the mixture of genomic material between individuals of different lineages (horizontal evolution). However, the standard evolutionary representation, the phylogenetic tree, doesn't represent faithfully the latter case.

To address this issue, in 2013, Chan, Carlsson and Rabadan[32] presented an evolutionary framework using TDA that extends beyond the limits of phylogenetic trees. Moreover, their method indicates the evolutionary scales where phylogenetic inference could be accurate.

In Figure 1.21, a metric space of pairwise genetic distances is calculated for a population of genomic sequences. Two genomes are joined by a line if their genetic distance is smaller than a chosen ε. Three genomes within ε of each other form a triangle, and so on: this procedure builds a simplicial complex. A one-dimensional cycle, highlighted in red, can be found at ε between 0.13 and 0.16 and it corresponds to a reticulate event.

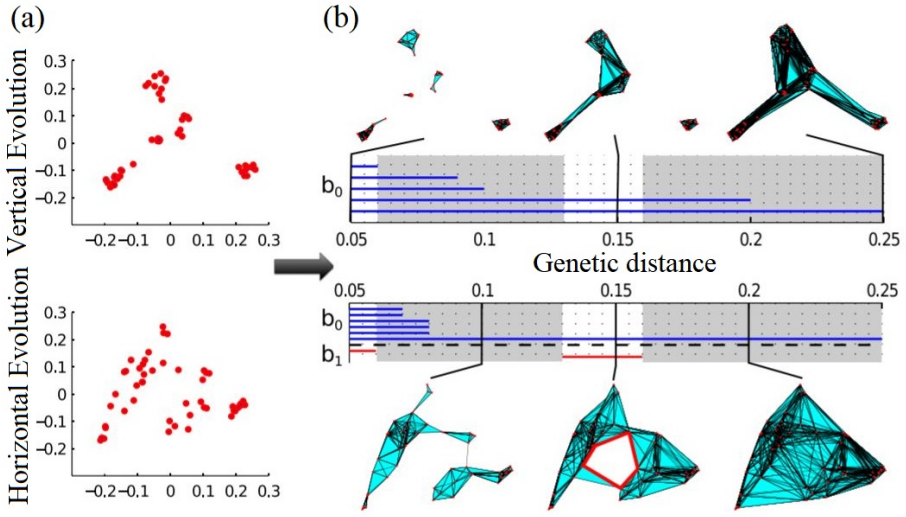


Figure 1.21: (a) Pairwise genetic distances. The resulting PCDs are show using PCA. (b) The filtration is derived and the homology groups are calculated at different scales. The resulting barcode is displayed. Image from [32].

1.2.4 Alternative Characterization of High-contrast Patches

The PH was used to find significant features hidden in a large data set of pixelated natural images (3x3 and 5x5 high-contrast patches). The subspace of linear and quadratic gradient patches forms a dense subset inside the space of all high-contrast patches and it was found to be topologically equivalent to the Klein bottle. See Figure 1.22.

This could lead to an efficient encoding of a large portion of a natural image: instead of using an "ad hoc" dictionary for approximating high-contrast patches, one can build such a dictionary in a systematic way by generating a uniform set samples from the ideal Klein bottle[35].

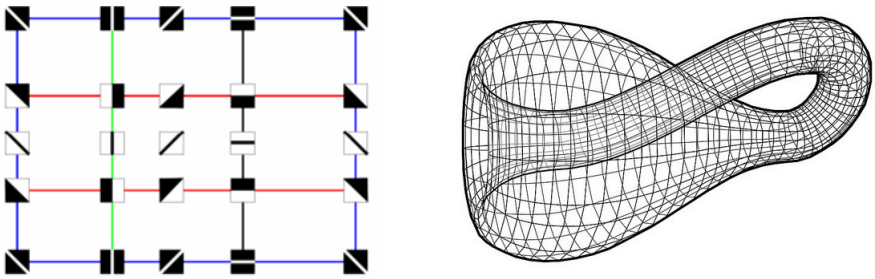


Figure 1.22: On the left, 3x3 patches parametrized by the Klein bottle. Image from [35]. On the right, a Klein bottle immersion in $\mathbb{R}^3$. Image from [15].

1.2.5 Finding Cosmic Voids and Filament Loops

In [17] and [51], PH is used to provide a substantial extension of available topological information about the structure of the Universe. See Figure 1.23.

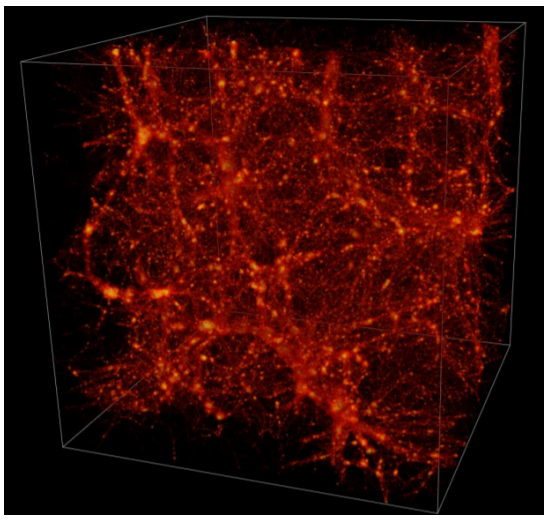


Figure 1.23: The Cosmic Web in an LCDM simulation, Image from [17].

While connencted components, loops and voids do not fully quantify topology, they extend the information beyond conventional cosmological studies of topology in terms of genus and Euler characteristic[17].

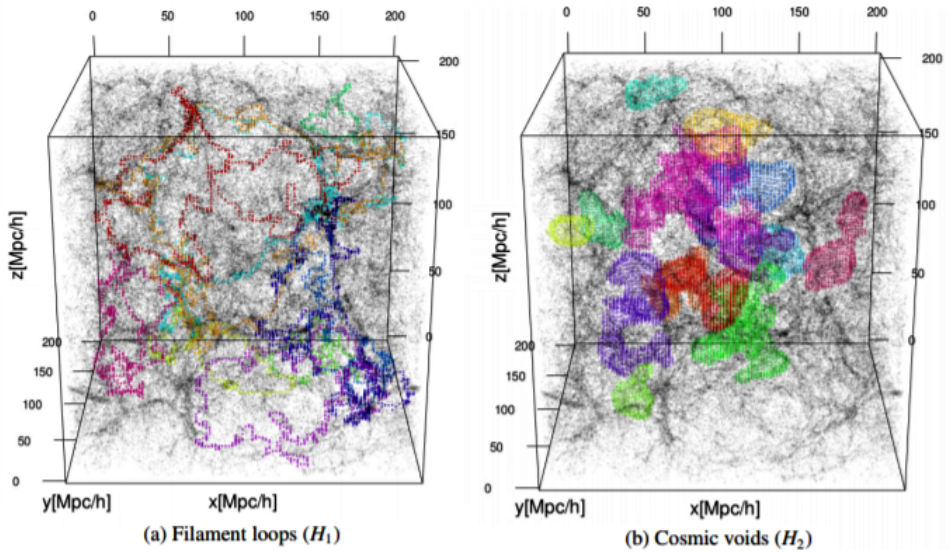


Figure 1.24: Filament loops (a) and voids (b) identified in the Libeskind et al. (2018) dataset[1] using SCHU. The most significant 10 filament loops (a) and the most significant 15 cosmic voids generators (b) are shown in different colors. All of their persistence values are less than 0.001. Images from [51].

2. Theoretical Background

Geometry deals with shapes, relative positions, sizes of figures and properties of space such as curvature. Topology studies the properties of space that are preserved under continuous deformations.

To understand how spaces agree and differ in shape, and so classifying them, we need to identify the intrinsic properties of spaces. In algebraic terms, this is translated into identifying certain elements of a given set as equivalent because of some of their features.

We will classify topological spaces up to homotopical equivalence.

In fact, spaces that are homotopy equivalent share many topological properties, in particular they have the same homology[48].

2.1 Fundamentals of Algebra

2.1.1 Equivalence Relations

Definition 2.1. Given a set $A \neq \varnothing$, an *equivalence relation* $\sim$ in A, is a binary relation between its elements such that:

- $x \sim x \quad \forall x \in A$ (reflexive property),
- $x \sim y \Rightarrow y \sim x \quad \forall x, y \in A$ (symmetrical property),
- $x \sim y \wedge y \sim z \Rightarrow x \sim z \quad \forall x, y, z \in A$ (transitive property).

A subset of A that contains all and only the elements equivalent to some element $x \in A$ is called *equivalence class* of x for the relation $\sim$.

Definition 2.2. Let A be a non-empty set and $\sim$ an equivalence relation defined on it. The *quotient set* of A for the relation $\sim$, $A/\sim$, is the set of equivalence classes obtained from A through $\sim$.

Definition 2.3. A *partition* of a set is a decomposition of the set into subsets such that every element of the set is in one and only one of them.

Observation 2.4. Any equivalence relation provides a partition of a set into equivalence classes, see Figure 2.1.

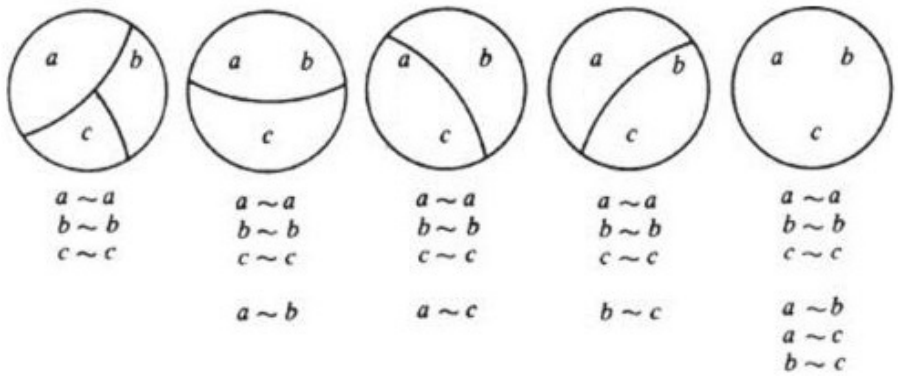


Figure 2.1: Example of equivalent relations and their partitions. If $A = \{a,b,c\}$, there are five ways of partitioning it. Under each partition is written the equivalence relation determined by it. Each partition of A determines and is determined by exactly one equivalence relation on A. Image from [109].

2.1.2 Homomorphisms

Definition 2.5. Given a set G equipped with a binary operation $* : G \times G \to G$, the couple $(G,*)$ is a *group* if and only if:

1. $\forall a,b,c \in G,\ a*(b*c) = (a*b)*c$ (associativity),
2. $\exists \mu \in G | \forall a \in G, \mu * a = a * \mu = a$ (existence of the *identity* μ),
3. $\forall a \in G, \exists a' \in G | a * a' = a' * a = \mu$ (existence of the *inverse* of a, a').

$(G,*)$ is *abelian* if, moreover, $\forall a,b \in G, a*b = b*a$.

Definition 2.6. Given a group $(G,*)$, $H \subset G$ is a *subgroup* of G if $(H,*)$ is a group.

Definition 2.7. A group $(G,*)$ is *cyclic* if $\exists g \in G | G =< g >= \{g^n | n \in \mathbb{Z}\}$.

Definition 2.8. Given two groups $(G,*)$ e $(H,\circ)$, a function $f : G \to H$ is a *homomorphism* if $\forall a,b \in G,\ f(a*b) = f(a) \circ f(b)$.

The purpose of defining a homomorphism is to create functions that preserve the algebraic structure of groups. Homomorphisms preserve the identity, the inverses, and the subgroups in the following sense.

Theorem 2.9. *Let ϕ be a homomorphism of a group G into a group H.*

- *If μ is the identity in G, then $\phi(\mu)$ is the identity in H.*
- *If $a \in G$, then $\phi(a') = \phi(a)^{-1}$*
- *If K is a subgroup of G, then $\phi(K)$ is a subgroup of H.*
- *If K is a subgroup of H, then $\phi(K)^{-1}$ is a subgroup of G.*

Definition 2.10. A bijective homomorphism is an *isomorphism*.

Two isomorphic groups differ only in the notation of their elements and are identical for all practical purposes.

Proposition 2.11. An isomorphism on any collection of groups is an equivalence relation on that collection of groups.

2.1.3 Structure Theorem

Definition 2.12. A set F with two binary operations defined on it, $+$ and $\cdot$, is a *field* if

- F is an abelian group under $+$ with 0 as additive identity.
- The non-zero elements are an abelian group under $\cdot$ with multiplicative identity.
- $\cdot$ is distributive over $+$.

Definition 2.13. A *vector space* on a field K is a set V with two operations $+ : V \times V \to V$ and $* : V \times V \to V$ such that:

- $(V,+)$ is an abelian group
- $a(\mathbf{u}+\mathbf{v}) = a\mathbf{u}+a\mathbf{v}, \forall a \in K,\ \forall \mathbf{u},\mathbf{v} \in V$
- $(ab)\mathbf{v} = a(b\mathbf{v}),\ \forall a,b \in K,\ \forall \mathbf{v} \in V$
- $(a+b)\mathbf{v} = a\mathbf{v}+b\mathbf{v}, \forall a,b \in K,\ \forall \mathbf{v} \in V$
- $1\mathbf{v} = \mathbf{v},\ \forall \mathbf{v} \in V$

Definition 2.14. The *dimension* of a vector space is the cardinality of one of its basis over its base field.

Example 2.15. The vector space $\mathbb{R}^3$ has $\{(1,0,0),(0,1,0),(0,0,1)\}$ as a basis, so $dim_{\mathbb{R}}(\mathbb{R}^3) = 3$. More generally, $dim_F(F^n) = n$ for any field F.

Definition 2.16. Let V be a vector space. A set $A \subset V$ is called *convex* if $\forall x,y \in A$ the segment that joins them, $\{(1-t)x+ty : t \in (0,1)\}$, is entirely contained in A.

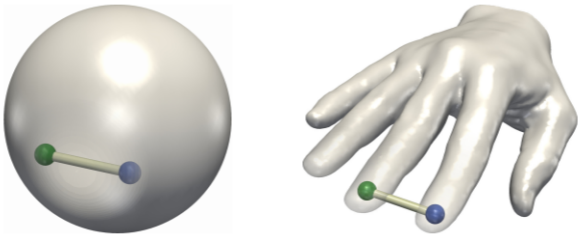


Figure 2.2: Examples of convex (left) and non-convex (right) 3-manifolds. Image from [118].

Definition 2.17. Given a set X with a binary operation $* : X \times X \to X$, the couple $(X,*)$ is a *monoid* if associativity and the existence of the identity hold.

Definition 2.18. Given a set R with two binary operations, $* : R \times R \to R$ and $@ : R \times R \to R$, the triplet $(R,*,@)$ is a *ring* if and only if:

- R is an abelian group under $*$,
- R is a monoid under $@$,
- $@$ is distributive with respect to addition.

R is *commutative* if $@$ is commutative.

Definition 2.19. For an arbitrary ring $(R, *, @)$, let $(R, *)$ be its additive group. A subset I is an *ideal* of R if it is an additive subgroup of R that satisfies the following conditions:

- $(I, *)$ is a subgroup of $(R, *)$
- $\forall x \in I, \forall r \in R : x@r, r@x \in I$

Definition 2.20. If every element of an ideal I can we written as $x = \sum_{k=1}^{n} a_k i_k$ where $a_k \in A$ and $\{i_k : k = 1, \ldots, n\}$ is a fixed subset of I, I is *finitely generated* and we write $I = (i_1, \ldots, i_n)$. If it is generated by only one element it is a *principal ideal*.

Definition 2.21. An *integral domain* is a non-zero commutative ring in which the product of any two non-zero elements is non-zero.

Definition 2.22. A *principal ideal domain* (PID) is an integral domain in which every ideal is principal.

Example 2.23. $\mathbb{Z}$, $\mathbb{Q}$ and $\mathbb{R}$ are PIDs.

Proposition 2.24. In a PID, any two elements x, y have a greatest common divisor, that can be get as a generator of $I = (x, y)$.

This property is needed by the Structure Theorem 2.34 that we will introduced. It allows to uniquely represent the PH with a barcode or a persistence diagram.

Definition 2.25. Given two abelian groups $(A, *)$ and $(B, @)$, their *direct sum* $A \oplus B$ is the cartesian product $A \times B$ with the operation $\cdot$ defined as $(a_1, b_1) \cdot (a_2, b_2) = (a_1 * a_2, b_1 @ b_2)$. For an infinite family of abelian groups $\{A_i\}_{i \in I}$, the direct sum is

$$\bigoplus_{i \in I} A_i = \{(a_i) \in \prod_{j \in I} A_j : a_i \textit{ is the identity element of } A_i \ \forall \textit{ but finitely many } i\}$$

Definition 2.26. A *graded ring* is a ring that is a direct sum of abelian groups R_i such that multiplication is defined as $R_i \otimes R_j \to R_{i+j}$. The elements in every R_i are called *homogeneous* and have degree i.

Example 2.27. Given a field F, the polynomial ring over it, $F[x]$, decomposes into $F[x] = \bigoplus_{i=0}^{\infty} x^i \cdot F$ where $x^i \cdot F = \{\sum_{i=0}^{\infty} a_i x^i : a_i \in F\}$. Moreover, the degree of the product of two monomials is the sum of the degrees of the factors, so $F[x]$ is a graded ring[121].

Definition 2.28. Given a ring R with multiplicative identity μ_R, a *left R-module* M is an abelian group $(M,+)$ with an operation $\cdot : R \times M \to M$ such that $\forall r,s \in R, \forall x,y \in M$:

- $r \cdot (x+y) = r \cdot x + r \cdot y$
- $(rs) \cdot x = r \cdot (s \cdot x)$
- $(r+s) \cdot x = r \cdot x + s \cdot x$
- $\mu_R \cdot x = x$.

Definition 2.29. A *graded module* is a left module M over a graded ring R such that $M = \bigoplus_{i \in \mathbb{N}_0} M_i$, and $R_i \otimes M_j \to M_{i+j}$.

Definition 2.30. A graded module M (ring R) is *non-negatively graded* if $M_i = 0$ ($R_i = 0$) for all $i < 0$.

Definition 2.31. Given a graded module M, $M(a)$ is the module M *shifted by a* steps so that $M(a)_d = M_{a+d}$.

We may grade $R[x]$ non-negatively with the grading proposed in [38] $(t^n) = t^n \cdot R[t], n \geq 0$.

Definition 2.32. The left R-module M is *finitely generated* if $\exists a_1, ..., a_n \in M$ such that $\forall x \in M, \exists r_1, ..., r_n \in R$ with $x = r_1 a_1 + ... + r_n a_n$.

Definition 2.33. A left R-module M is *cyclic* if $\exists x \in M | M = (x)$.

The Structure Theorem for finitely generated modules over a PID intuitively states that finitely generated modules over a PID can be uniquely decomposed similarly to integers through prime factorization.

Theorem 2.34 (Structure Theorem[38])**.** *If D is a PID, every finitely generated D-module decomposes uniquely as*

$$D = D^{\beta} \oplus (\bigoplus_{i=1}^{m} D/d_i D),$$

for $d_i \in D$: $d_i/d_{i+1} \in \mathbb{Z}$ and $\beta, m \in \mathbb{Z}$. Similarly, every graded module M over a graded PID D decomposes uniquely as

$$M = (\bigoplus_{i=1}^{n} \Omega_{\alpha_i} D) \oplus (\bigoplus_{i=1}^{m} \Omega_{\gamma_i} D/d_j D)$$

where $d_j \in D$ are homogeneous elements so that $d_j/d_{j+1}, \alpha_i, \gamma_i \in \mathbb{Z}$ and Ω_α denotes an α-shift upward in grading, and $m, n \in \mathbb{Z}$.

In both cases the theorem decomposes the structures into a left *free* submodule that includes generators able to build an infinite number of elements, in particular we have a vector space of dimension β, and a right *torsional* portion whose generators may build a finite number of elements. For example, if $D = \mathbb{Z}$, $\mathbb{Z}/3\mathbb{Z} = \mathbb{Z}_3$ is a generator that can build three elements. The torsional elements d_i are homogeneous.

2.2 Fundamentals of Topology

2.2.1 Topological Spaces

The notion of topological space represents a very general concept of space having a notion of "closeness" between elements defined in the weakest possible way.

Definition 2.35. Given a set $X \neq \varnothing$, a set τ is a *topology* on X if and only if:

1. $\tau \subseteq \mathscr{P}(\mathscr{X})$ (that is, τ is a set of subset of X),
2. $\varnothing, X \in \tau$,
3. τ is closed with respect to arbitrary union,
4. τ it is closed with respect to finite intersection.
   Elements of τ are called *open sets* in τ, while elements in X that are not in τ are called *closed set* in τ.

Intuitively, open sets are subsets of topological spaces which do not contain their boundaries[118]. For example in $\mathbb{R}$, $(-\infty, 0) \cup (1, +\infty)$ and $[0, 1]$ are respectively open and closed sets.

Definition 2.36. Given a set $X \neq \varnothing$ and a topology on it τ, (X, τ) is a *topological space*.

Definition 2.37. Let (X, τ) be a topological space, and let $Y \subseteq X$. The *subspace topology* τ_Y on Y is $\tau_Y = \{U \cap Y : U \in \tau\}$. τ_Y is the topology "induced by" or "inherited from" τ.

Definition 2.38. Let (X, τ) be a topological space, and let $\sim$ be an equivalence relation on X. The corresponding *quotient topological space* is given by the topological space $(X/\sim, \tau_\sim)$ where $\tau_\sim$ is the *quotient topology* where the open sets are defined to be those sets of equivalence classes whose unions are open sets in X.

Example 2.39. Given a topological space X and points $x, y \in X$, we can "glue" them with the equivalence relation $\sim$ such that $(a \sim b) \Leftrightarrow (a = b \vee (a = x \wedge b = y) \vee (a = y \wedge b = x))$.

Definition 2.40. Given a $x \in (X, \tau)$, a set A is called *neighbourhood* of x if $A \subseteq X$ and A contains an open set containing x.

Observation 2.41. The concept of neighbourhood of x represents intuitively a set of points "close" or "similar" to x.

Example 2.42. Given a set X we can always define the *discrete topology* $\mathscr{D} = \mathscr{P}(X)$. It is the *finest* topology that can be given on a set: it defines all subsets as open sets. In particular, each singleton is an open set in the discrete topology, meaning that each of them is isolated from the others.

Example 2.43. The *trivial topology* is the topology with the least possible number of open sets, namely the empty set and the entire space. All points here are closed because they all are sharing the same neighbourhood.

Example 2.44. the *Euclidean topology* is the natural topology induced on Euclidean $n-$space $\mathbb{R}^n$ by the Euclidean metric. The open sets of the Euclidean topology on $\mathbb{R}^n$ are given by (arbitrary) unions of the open balls $B_r(p)$ defined as $B_r(p) := \{x \in \mathbb{R}^n \mid d(p,x) < r\}\ \forall r > 0,\ \forall p \in \mathbb{R}^n$, where d is the Euclidean metric.

Even if we are used to see $\mathbb{R}$ intuitively as a straight line, $\mathbb{R}^2$ as a plane, etc., these representations are valid only in Euclidean topology. In fact the shape of a set is determined by its topology.

Example 2.45. $\mathbb{R}$ with the discrete topology can be seen as a cloud of separate and unordered points, while with the trivial one as a single "big" point because all real numbers are neighbors.

Definition 2.46. A topological space X is *compact* if for every collection C of open subsets of X such that $X = \bigcup_{x \in C} x$, there is a finite subset F of C such that $X = \bigcup_{x \in F} x$.

Definition 2.47. A topological space X is *connected* if for any two points of X there exists a path between them on X.

Definition 2.48. The maximally connected subsets of a topological space are its *connected components*.

Example 2.49. Euclidean space is connected while any discrete space of size more than one is not connected.

2.2.2 Homeomorphisms

A homeomorphism is a function between topological spaces that models the intuitive idea of deformation without tearing, overlapping or gluing.

Definition 2.50. Given two topological spaces X and Y, a function $f : X \to Y$ is a *homeomorphism* among them if and only if f it is bijective, continuous and its inverse is continuous.

Definition 2.51. Two topological spaces X and Y are *homeomorphic*, $X \simeq Y$, if and only if there exists a homeomorphism from one to another (or equivalently if they have the same topological properties).

General Topology studies the topological properties of topological spaces. These features are the ones preserved by homeomorphisms, so they are also called *topological invariants*.

Observation 2.52. Intuitively, Topology studies how things are connected, not how they look. So, it allows to define concepts such as continuity, compactness, connection and closeness even outside of $\mathbb{R}^n, n \in \mathbb{N}$.

Example 2.53. Given $X = (-1,1)$ and $f : X \to \mathbb{R}, f(x) := tan(\frac{\pi x}{2}), \forall x \in X$, f is a homeomorphism. Note that X and $\mathbb{R}$ have different lengths. Moreover X is bounded and $\mathbb{R}$ is not. Therefore length and boundedness are not topological properties.

Homeomorphism is the most fundamental notion of topological equivalence, for example a doughnut and a cup are homeomorphic because we can derive one from the other as shown in Figure 2.3.

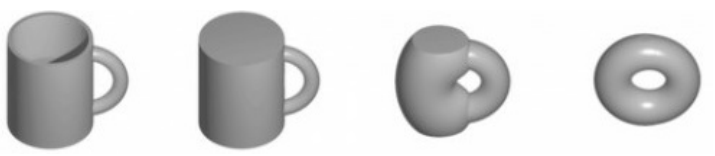


Figure 2.3: A cup deformed into a doughnut without gluing or cutting. Image from [73].

Example 2.54. Let $X_\varepsilon = ([0,1], \varepsilon)$ where ε represents the 1D-Euclidean topology. Let $\sim$ be the equivalence relation on X_ε such that $0 \sim 1$. Intuitively, we get a circle, but to prove that we should prove that $X_\varepsilon / \sim$ is homeomorphic to one. See Figure 2.4.

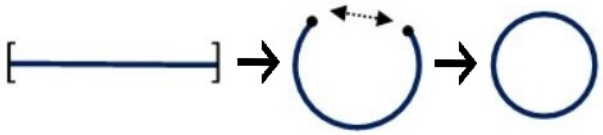


Figure 2.4: Example of equivalent relation $\sim$: $0 \sim 1$ *applied to the topological space* $([0,1], \varepsilon)$.

We now introduce *manifolds* to help the reader to understand more deeply topological classifications.

2.2.3 Manifolds and Betti Numbers

Definition 2.55. A topological space M is a *d-manifold* if every element $m \in M$ has an open neighborhood N homeomorphic to an open Euclidean d-dimensional ball.

Example 2.56. An intuitive description of a d-manifold is given by a curved space. It has the structure of an Euclidean space of dimension d locally, while globally has a more complicated structure. Euclidean spaces are examples of manifolds[118]. See Figure 2.5.

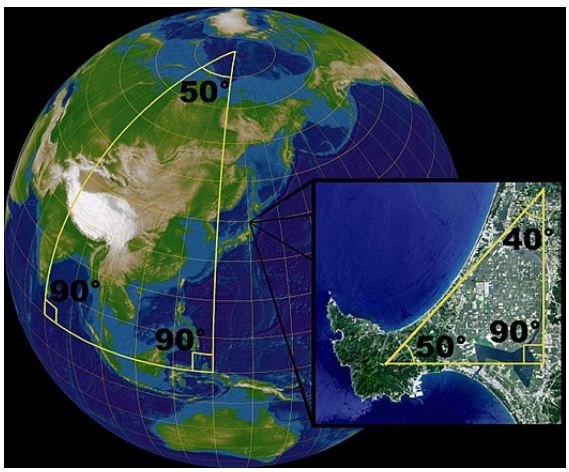


Figure 2.5: Locally the earth's surface resembles a plane, so it is a 2-manifolds. However, this similarity does not preserve the distance between the points, since the sphere has a different curvature. We can see as the curvature affects the sum of the internal angles of a triangle: in the plane this sum is always 180°, while on a sphere it is always greater. Image from [111].

Definition 2.57. We'll refer to 1-manifolds as *curves* and to 2-manifolds as *surfaces*.

Example 2.58. Any discrete space is a 0-manifold while $\mathbb{R}^n$ is a n-manifold.

Given a tridimensional PCD, the non-manifold regions can be identified by local dimension estimation[101] and splitted into dimensional manifold regions, as shown in Figure 2.6.

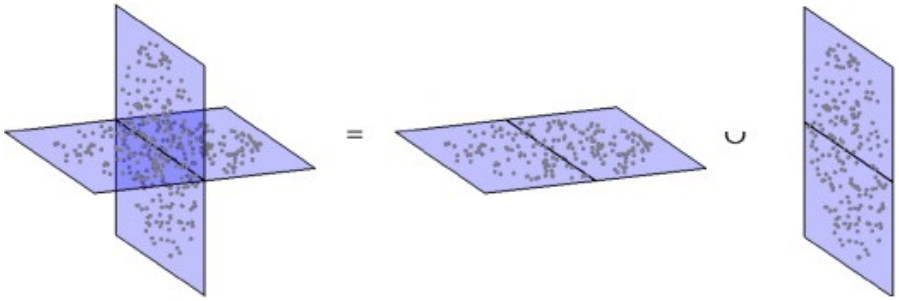
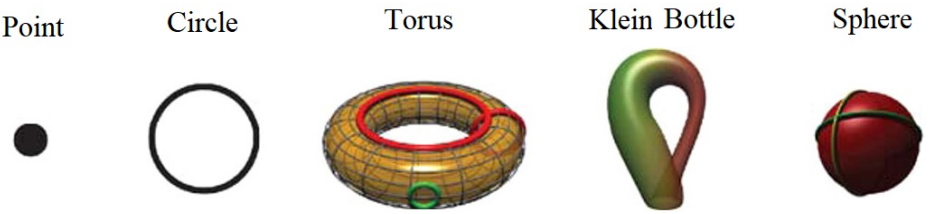


Figure 2.6: The decomposition of a 3D non-manifold neighborhood of the type accepted by the algorithm described in [101] into two 2D manifold neighborhoods. Image from [101].

Important examples of topological invariants are the number of independent components, rings and cavities called *Betti-0, Betti-1* and *Betti-2*, respectively. See Figure 2.7.



	Point	Circle	Torus	Klein Bottle	Sphere
Betti-0	1	1	1	1	1
Betti-1	0	1	2	2	0
Betti-2	0	0	1	1	1
Betti-n, n>2	0	0	0	0	0

Figure 2.7: Betti numbers of some shapes. For the torus, two auxiliary rings are added to explain Betti-1= 2. Image rearranged from [34].

From Betti numbers another important invariant can be derived: the *Euler characteristic*.

Definition 2.59. Given a topological space X and its Betti numbers β_i, its Euler characteristic is

$$\chi(X) = \sum_{i=0}^{\infty} (-1)^i \beta_i$$

Betti numbers can bring very useful insights for data analysis in their own right. However, in some cases they can contribute also to infer information about the global structure of the

data. For example, Betti numbers were shown to completely classify compact connected 2-manifold without boundary.

In [53], a dataset of more than a million cyclo-octane conformations were analysed and the obtained Betti numbers where (1,1,2). Since the cyclo-octane surface is non-manifold, the Betti numbers were uninformative.

Although, using a triangulation of the conformation space, the researcher decomposed the object into the two components distinguishable in Figure 2.8: the outer sphere and the enclosed hourglass. Both of these objects were found to be compact connected surfaces without boundary, each sharing points on the two intersection rings.

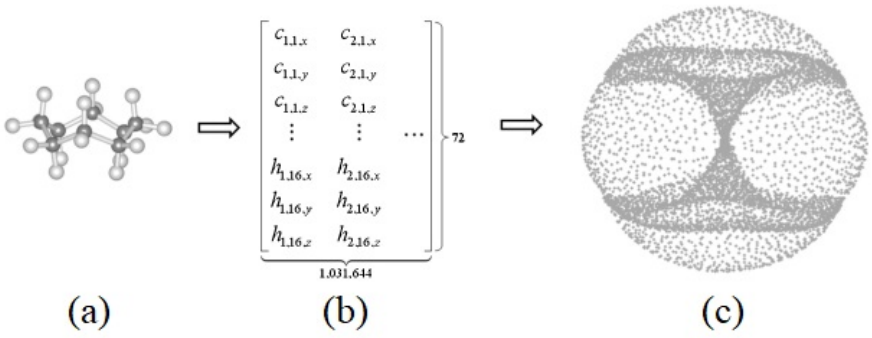


Figure 2.8: An example of conformation of cyclo-octane represented by the 3D coordinates of its atoms(a). The coordinates are concatenated into vectors and shown as columns of a data matrix (b). In (c), the Isomap method is used to obtain a lower dimensional visualization of the data. Image rearranged from [53].

Unsurprisingly, the Betti numbers of the spherical component were (1,0,1). Instead, the Betti numbers of the hourglass were (1,1,0): the Klein bottle ones. This object cannot be embedded in less than four dimensions and this confirmed that 5D is necessary to fully capture cyclo-octane conformation space.

TDA makes very few assumptions about the data and the goal is not to faithfully reconstruct the data or to fit the data to a model but to provide unbiased summaries of the geometric/topological structure of the data.

Nevertheless, for the sake of completeness, we report an important result on the classification of 2d-manifolds and in Theorem 3.25 we'll give a full characterization of the compact surfaces in terms of orientability and Euler characteristic.

Theorem 2.60 ([18])**.** *Every compact surface is homeomorphic to one and only one of the following:*

- *the sphere,* $\mathbb{S}^2$*,*
- *the arbitraty connected sum of torus,* $\mathbb{T}^2\#\ldots\#\mathbb{T}^2$*,*
- *the arbitraty connected sum of real projective plane,* $\mathbb{RP}^2\#\ldots\#\mathbb{RP}^2$*.*

2.2.4 Homotopy

A more flexible notion of equivalence is the homotopy one. Intuitively, two continuous functions, defined from a topological space to another, are *homotopic* if one of them can be continuously deformed into the other, see Figure 2.9.

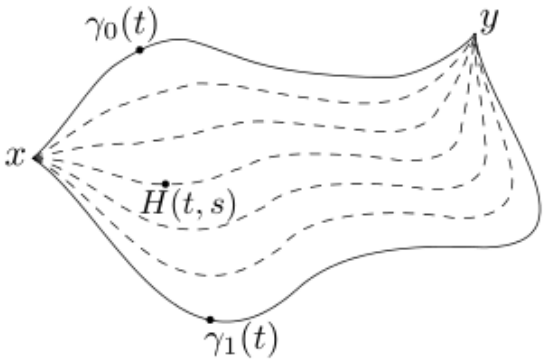


Figure 2.9: Representation of a homotopy H between two curves γ_0 *e* γ_1*. Image from [2].*

The notion of homotopy equivalence is weaker than the notion of homeomorphism: all homeomorphic spaces are also homotopy equivalent but the converse is not necessarily true. However, spaces that are homotopy equivalent share the same homology.

Definition 2.61. Given two topological spaces X and Y and two continuous functions f and g from X to Y, an *homotopy* from f to g is defined to be a continuous function $H : X \times [0,1] \to Y$ such that, if $x \in X$ then $H(x,0) = f(x)$ and $H(x,1) = g(x)$.

Example 2.62. If $f, g : \mathbb{R} \to \mathbb{R}^2$ such that $f(x) = (x, x^3)$ and $g(x) = (x, e^x)$, then the map $H : \mathbb{R} \times [0,1] \to \mathbb{R}^2$ given by $H(x,t) = (x, (1-t)x^3 + te^x)$ is a homotopy between them.

Definition 2.63. Two topological spaces X and Y are *homotopy equivalent* if there exist

continuous maps $f : X \to Y$ and $g : Y \to X$ such that $g \circ f$ is homotopic to the identity map id_X and $f \circ g$ is homotopic to id_Y.

See Figure 2.4 to visualize an example of homotopy equivalence.

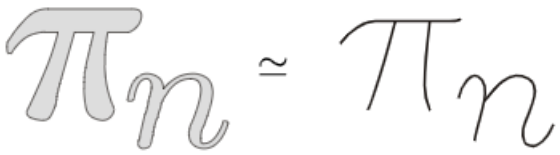


Figure 2.10: These two sets are homotopy equivalent. Image from [54].

Example 2.64. The Möbius strip and an untwisted strip are homotopy equivalent since we can deform both continuously to a circle, but they are not homeomorphic, see Figure 2.11.

Figure 2.11: Three homotopy equivalent shapes: a Möbius strip, a circle and an untwisted strip. Image rearranged from [100].

Definition 2.65. A space is said to be *contractible* if it's homotopy equivalent to a point.

As mentioned before, homotopy equivalence is used to classify topological spaces.

The homotopy equivalent items are collected into equivalence classes, called *homotopy classes*. They form a group, called the *n-th homotopy group*, $\pi_n(X)$, of the given space X.

Definition 2.66. In the n-sphere S^n we choose a base point a. For a space X with base point b, we define $\pi_n(X)$ to be the set of homotopy classes of maps $f : S^n \to X$ such that a is mapped into b.

The first and simplest homotopy group is the *fundamental group*, which counts how many loops there are in a space. See Figures 2.12 and 2.13.

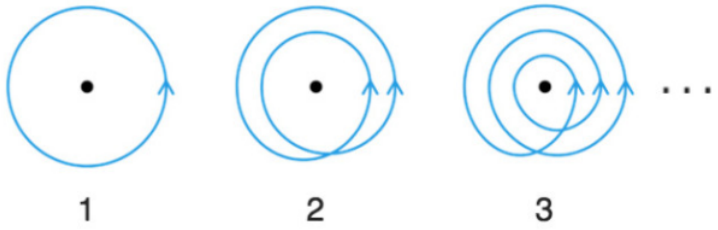


Figure 2.12: $\pi_1(S^1) = \mathbb{Z}$. We can wrapping a band around a rod as many time as we want. The wrappings with opposite directions cancel out each other. $\pi_1(S^1)$ is an infinite cyclic group, and it is isomorphic to the group $\mathbb{Z}$ under addition: a homotopy class is identified with an integer by counting the number of times a mapping in the homotopy class wraps the circle. Image from [86].

Figure 2.13: $\pi_1(S^2) = \mathbf{0}$. Any continuous mapping from a circle to a sphere can be deformed into one-point with continuity. So its homotopy class has only one element, the identity element and $\pi_1(S^2)$ can be identified with the subgroup of $\mathbb{Z}$ having only of the zero, $\mathbf{0}$. Image from [94].

Homotopy groups are algebraic objects that intuitively measures the amount of "n-dimensional holes" of a space.

2.2.5 Metric Spaces

A metric space is a set of elements, called *points*, in which a distance between them is defined. It represents a particular type of topological space.

Definition 2.67. A *distance* (o *metric*), on a not empty set X, is any function $d : X \times X \to \mathbb{R}$ such that $\forall x, y, z \in X$:

- $d(x,y) \geq 0$
- $d(x,y) = 0 \iff x = y$
- $d(x,y) = d(y,x)$ (symmetry)
- $d(x,y) \leq d(x,z) + d(z,y)$ (triangular inequality)

Definition 2.68. A *metric space* is a mathematical structure consisting of a pair (X,d) of elements, where X is a non-empty set and d is a metric on X.

Example 2.69. The Euclidean metric in the plane is our intuitive distance function on flat surfaces. To measure a distance between two points, we use the distance function $d_2(x,y) = \sqrt{(x_1-y_1)^2+(x_2-y_2)^2}$. There are some less intuitive metrics that we can use. We can define the Manhattan metric on the plane with the distance $d_1(x,y) = |x_1-y_1| + |x_2-y_2|$. While in the 'infinity' metric we use distance $d_\infty(x,y) = Max\{|x_1-y_1|, |x_2-y_2|\}$. By *circle* we mean the set of all points lying at a given distance from a fixed point. In Figure 2.14 we can see that a circle is a diamond with d_1 and a square with d_∞. See Figure 2.15.

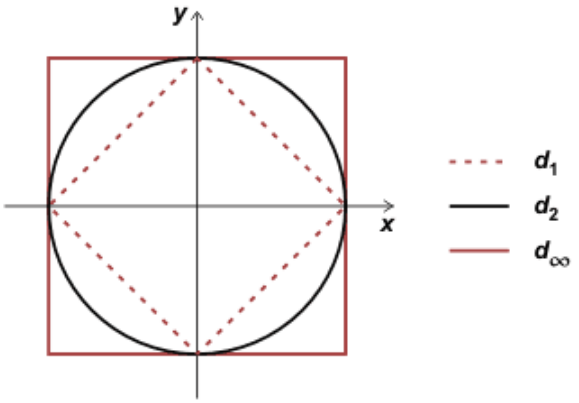


Figure 2.14: The representation of a circle using $(\mathbb{R}^2, d_2)$, $(\mathbb{R}^2, d_1)$ and $(\mathbb{R}^2, d_\infty)$ metrics.

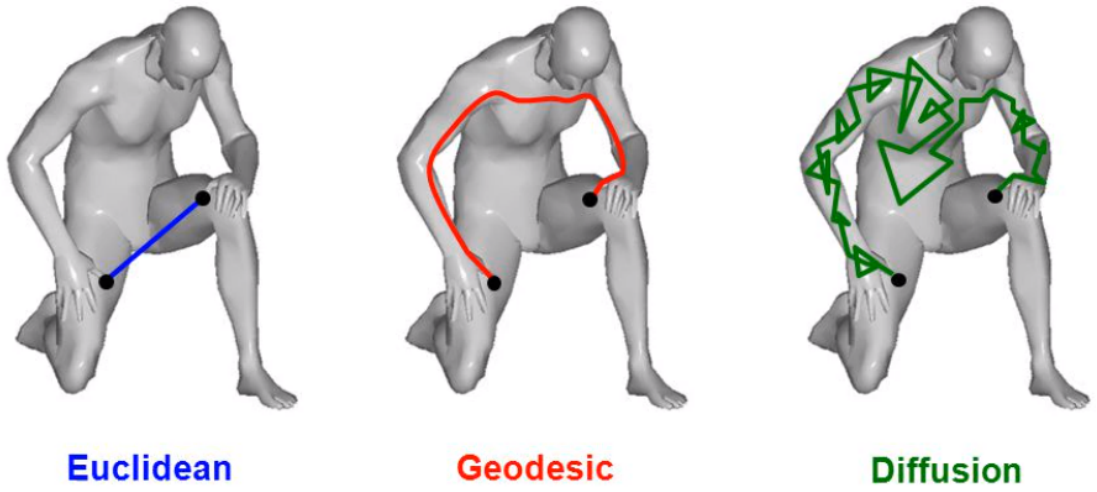


Figure 2.15: Example of distances beetwen two points. Image from [19].

Definition 2.70. A *finite* metric space is a metric space having a finite number of points, that is a PCD.

Proposition 2.71. Any metric space (X,d) is compact if and only if it is complete and totally bounded.

Observation 2.72. Any PCD is compact.

Definition 2.73. Let X and Y be metric spaces with metrics d_X and d_Y. A map $f : X \to Y$ is an *isometry* if $\forall a,b \in X, d_Y(f(a), f(b)) = d_X(a,b)$. X and Y are *isometric* if a bijective isometry exists between them.

Example 2.74. Any rotation, translation and reflection is an isometry on Euclidean spaces.

2.2.6 Hausdorff distance

The *Hausdorff distance* provides a convenient way to quantify the proximity between data sets issued from the same metric space.

Definition 2.75. Given a topological space X, it is *Hausdorff* if for every couple of points $x, y \in X$ there exists a neighborhood U of x and a neighborhood V of y such that U and V are disjoint.

Observation 2.76. Any metric space is Hausdorff.

Definition 2.77. Let X and Y be two non-empty subsets of a metric space (M,d). We define their Hausdorff distance $d_H(X,Y)$ by

$$d_{\mathrm{H}}(X,Y) = \max\{ \sup_{x\in X} \inf_{y\in Y} d(x,y), \sup_{y\in Y} \inf_{x\in X} d(x,y) \},$$

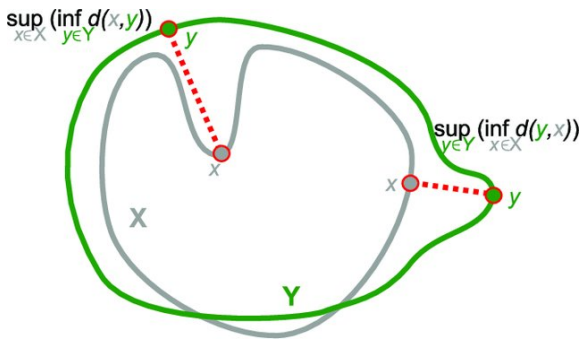


Figure 2.16: Components of the calculation of the Hausdorff distance between X and Y. Image from [108].

See Figure 2.16.

To compare datasets not sampled from the same ambient space, the notion of Hausdorff distance can be generalized to the Gromov-Hausdorff one.

Thanks to it, any pair of datasets issued from compact metric spaces can be compared. We will use these concepts in the study of the stability of the PH method.

Definition 2.78. Given two metric spaces M_1 and M_2, the *Gromov-Haudoff distance* $d_{GH}(M_1, M_2)$ is the infimum of the $r \geq 0$ such that there exists a metric space (M, ρ) isometric to M_1 and M_2 and such that $d_H(M_1, M_2) \leq r$.

The Gromov-Hausdorff distance measures how far two metric spaces are from being isometric. See Figure 2.17.

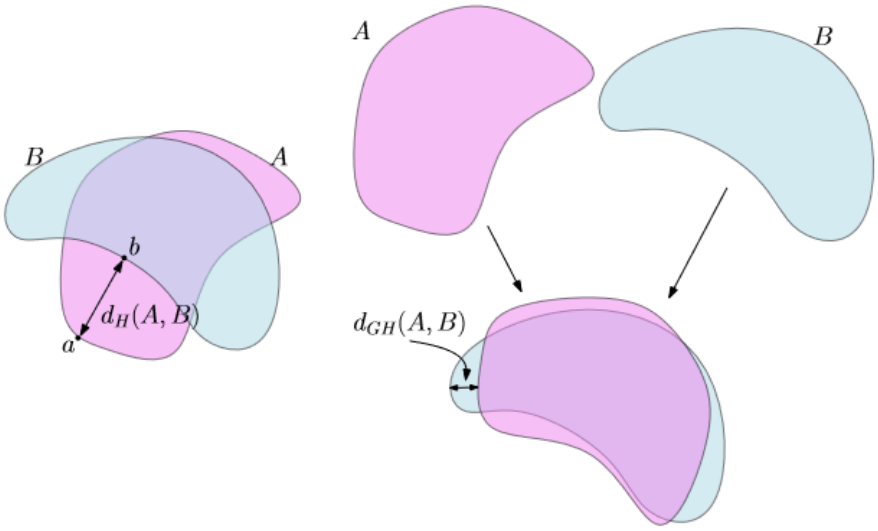


Figure 2.17: On the right the Hausdorff distance between two subsets A and B of the plane. On the left the Gromov-Hausdorff distance between A and B. Rotation is an isometric embedding of A in the plane, so A can be rotated to reduce its Hausdoff distance to B. Image from [48].

Proposition 2.79. Given two metric spaces M_1 and M_2 that are subspaces of the same metric space, $d_{GH}(M_1, M_2) \leq d_H(M_1, M_2)$.

After this recap of the basic algebraic topology tools that will be used, we move to the first step of TDA: define simplicial complexes.

3. Topological Data Completion

3.1 From Simplexes to Filtrations

Connecting similar points by edges pairs leads to the concept of neighboring graph from which the connectivity of the data can be studied using clustering algorithms. To go beyond connectivity, TDA builds higher dimensional counterparts of neighboring graphs. This is achieved by connecting $(k+1)$-uple of nearby points. The resulting objects, called simplicial complexes, allow identifying topological features such as cycles and voids.

The torus and the Klein bottle can be obtained from a square by identifying opposite edges as indicated in Figure 3.1. Cutting a square along a diagonal produces two triangles, so each of these surfaces can also be built from two triangles by identifying their edges in pairs.

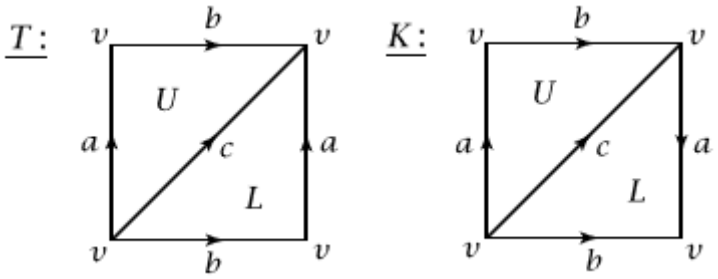


Figure 3.1: Representation on a torus (left) and a Klein bottle (right) using squares. Image from [81].

The idea of a simplicial complex is to generalize structures like these to any number of dimensions. We think of an $n-simplex$ as an n-dimensional triangle, and we can "triangulate" a space by gluing a bunch of these together.

3.1.1 Simplexes

We denote the vertices of the simplex as p_i, and the simplex as $[p_0, ..., p_k]$.

Definition 3.1. Let $P = p_0, p_1, ..., p_k \subseteq \mathbb{R}^d$.

- A *linear combination* is $x = \sum_{i=0}^{k} \lambda_i p_i$, for some $\lambda_i \in \mathbb{R}$.
- An *affine combination* is a linear combination with $\sum_{i=0}^{k} \lambda_i = 1$.
- A *convex combination* is an affine combination with $\lambda_i \geq 0, \forall i$.
- The set of all convex combinations is the *convex hull*.
- A set P is *affinely (linearly) independent* if no one of its points is affine (linear) combinations of the other points.

Definition 3.2. A *k-simplex* is the convex hull of $k+1$ affinely independent points called *vertices*.

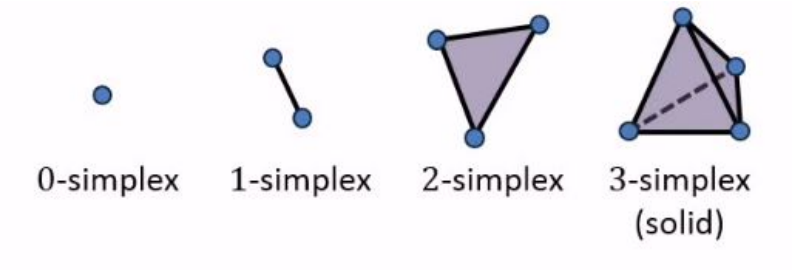


Figure 3.2: k-simplices, $\forall k : 0 \leq k \leq 3$. Image from [125].

Summarizing, given $p_0, ..., p_k \in \mathbb{R}^k$ such that $p_1 - p_0, ..., p_k - p_0$ are linearly independent, a k-simplex determined by them is

$$\sigma := \{\lambda_0 p_0 + ... + \lambda_k p_k \mid \lambda_i \geq 0, 0 \leq i \leq k, \sum_{i=0}^{k} \lambda_i = 1\}.$$

Observation 3.3. A k-simplex is a k-dimensional polytope which is the convex hull of its $k+1$ vertices[112].

Definition 3.4. Given a set of simplexes K, their union is a subset of $\mathbb{R}^d$ called *underlying space of K* that inherits from the topology of $\mathbb{R}^d$.

Definition 3.5. Let σ be a k-simplex defined by $P = \{p_0, p_1, ..., p_k\}$. A simplex τ defined

by $T \subset P$ is a *face* of σ, $\sigma > \tau$.

Observation 3.6. A k-simplex has $\sum_{l=-1}^{k} \binom{k+1}{g+1} = 2^{k+g}$ faces and in particular $\binom{k+1}{g+1}$ faces of dimension g.

Definition 3.7. The *vertices* of P are the zero-simplices in P.

Example 3.8. Given the 4 vertices of a tetrahedron, there are 4 different subsets composed of 3 vertices each that are 4 triangular faces.

There are many tipes of complexes. For example digital images have a cubical structure, given by the pixels (in 2D) or voxels (in 3D). Therefore, one approach to studying digital images uses combinatorial structures called *cubical complexes*[74].

A generalization of the simplicial complex is the *polytopal complex* or *cellular complex*. It consists of a collection X of convex polytopes in some Euclidean space $\mathbb{R}^d$ such that every face of a polytope in X is in X and the intersection of any two polytopes in X is a face of both[60]. An n-dimensional polytope in X is called an *n-cell*. However, we will discuss the simplicial complex that is more common in literature.

3.1.2 Simplicial Complexes

Definition 3.9. A *geometric simplicial complex* P is a finite collection of non-empty simplices such that every face τ of every simplex of P is a simplex of P, and if two complexes intersect, this occurs on common a face.

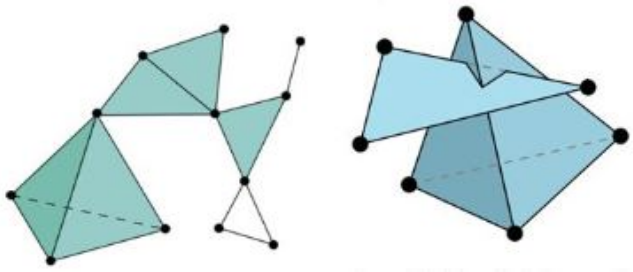


Figure 3.3: On the left, we have an example of a simplicial complex, on the right some simplices that do not intersect properly to build a simplicial complex. Image from [116].

Definition 3.10. The *dimension* of P is $dim(P) = max\{dim(\sigma)|\sigma \in P\}$, where $dim(\sigma)$ equals the number of elements in σ.

Observation 3.11. The simplicial complexes of dimension 1 are graphs.

Simplicial complexes can be defined without using any geometry, although it may not seem. We will present this definition next, as it displays the separation between topology and geometry.

Definition 3.12. An *abstract simplicial complex* is a set P with a collection S of its subsets such that $\forall v \in P, \{v\} \in S$ and $\forall \sigma \in S$, if $\tau \subseteq \sigma$, then $\tau \in S$. The sets $\{v\}$ are the *vertices* of P.

Definition 3.13. σ is a simplex of dimension k if $|\sigma| = k+1$. If $\tau \subseteq \sigma$, τ is a face of σ.

Definition 3.14. The *maximal faces* or *facets* of a complex are those faces that are not subsets of any other faces.

The structure of a simplicial complex can be fully specified by the list of its facets.

Definition 3.15. Given an abstract simplicial complex P, suppose $\tau, \sigma \in P$ such that $\tau \subset \sigma$ and in particular $\dim(\tau) < \dim(\sigma)$. If σ is a maximal face of P and no other maximal face of P contains τ, τ is a *free face*.

We now relate this abstract set-theoretic definition to the geometric one by extracting the combinatorial structure of a simplicial complex.

Definition 3.16. Let P be a simplicial complex with vertices V and let $\mathscr{P}$ be the collection of all subsets $\{v_0, v_1, ..., v_k\}$ of V such that $v_0, v_1, ..., v_k$ span a simplex of P. $\mathscr{P}$ is the *vertex scheme* of P.

Definition 3.17. Given two abstract simplicial complexes P_1 and P_2 having vertex sets V_1 and V_2 respectively, an *isomorphism* between them is a bijection $\phi : V_1 \to V_2$ such that P_1 and P_2 are the same except for a relabelling of their vertices by ϕ and ϕ^{-1}.

Theorem 3.18. *For every abstract complex P there exist a geometric simplicial complex K whose vertex scheme is isomorphic to P. K is a geometric realization of P. Two simplicial complexes are isomorphic if and only if their vertex schemes are isomorphic as abstract simplicial complexes.*

The geometric realization of P is uniquely determined up to an isomorphism. So, abstract

simplicial complexes can be seen as topological spaces and geometric complexes can as geometric realizations of their combinatorial structure.

Example 3.19. Consider the simplicial complex $\{\{a\},\{b\},\{c\},\{d\},\{e\},\{a,b\},\{a,c\},\{a,d\},\{b,c\},\{c,d\},\{a,b,c\}\}$. Its geometric realization is the subset of $\mathbb{R}^2$ is displayed in Figure 3.4.

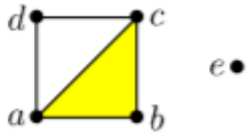


Figure 3.4: The geometric realization of the simplicial complex in $\mathbb{R}^2$. Image from [74].

Simplicial complexes are at the same time combinatorial objects (well-suited for effective computations) and topological spaces (from which topological properties can be inferred).

Definition 3.20. Given a topological space X, a simplicial complex K homeomorphic to it, together with a homeomorphism $h : K \to X$, is a *triangulation* of X.

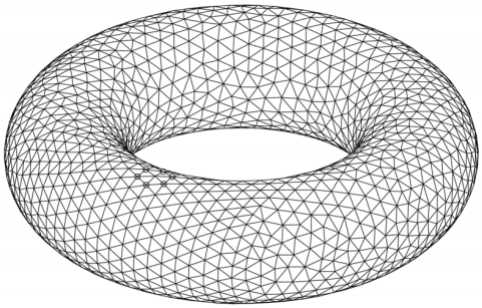


Figure 3.5: A triangulated torus. Image from [85].

For purposes of homology it will be important to keep track of the order of the vertices of a simplex.

Definition 3.21. Let P be a simplicial complex. An *orientation* of a k-simplex $\sigma \in P$, $\sigma = [v_0, v_1, ..., v_k]$, $v_i \in P$, is an equivalence class of orderings of the vertices of σ, where $(v_0, v_1, ..., v_k) \sim (v_{\tau(0)}, v_{\tau(1)}, ..., v_{\tau(k)})$ are equivalent orderings if the parity of the permutation τ is even.

Orientations may be shown graphically using arrows, as shown in Figure 3.6.

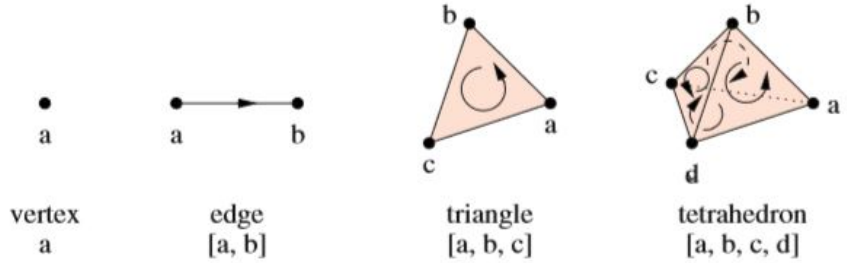


Figure 3.6: k-simplices, $0 \geq k \geq 3$. The orientation on the tetrahedron is shown on its faces. Image from [38].

If we delete one of the $n+1$ vertices of an n-simplex $[v_0, ..., v_n]$, then the remaining n vertices span an $(n-1)$-simplex. According to letterature, we adopt the convention that the vertices of any subsimplex spanned by a subset of the vertices, will be ordered according to their order in the larger simplex.

Definition 3.22. Two k-simplices that share a $(k-1)$-face are consistently oriented if they induce different orientations on it.

Theorem 3.23 ([18])**.** *Every surface has a triangulation. Each facet of a triangulation of a surface is of dimension 2.*

Definition 3.24. A surface is *orientable* if all the 2-simplices of a triangulation of it can be consistently oriented. Otherwise it is *non-orientable*.

We have can provide a full characterization of the compact surfaces in terms of orientability and Euler characteristic.

Theorem 3.25 ([18])**.** *If a compact surface X is orientable, it is homeomorphic to*

- $\mathbb{S}^2$ *if* $\chi(X) = 2$,
- $\mathbb{T}^2 \# \ldots \# \mathbb{T}^2$ *where there are g summands if* $\chi(X) = 2-2g \neq 2$.

If a compact surface X is not orientable, it is homeomorphic to $\mathbb{RP}^2 \# \ldots \# \mathbb{RP}^2$ *where there are g summands if* $\chi(X) = 2-g$.

Getting back to the main topic, there exist many ways to build simplicial complexes from a dataset, or more generally from a topological or metric space.

To be a useful, a simplicial complex has to satisfy some properties: intuitively, its homology has to approximate the one of the space we want to study.

See Figure 3.7.

Figure 3.7: Three possible complexes build from the sample produced by a sensor observing an annulus. Only the first complex provides a reasonable approximation of it. Image from [121].

For example, for the Čech complex, these properties are guaranteed by the Nerve Theorem. In Table 3.1 a summarization of some simplicial complexes is shown.

Complex	Size	Justification
Čech	$2^{O(N)}$	Nerve Theorem
Vietoris-Rips (VR)	$2^{O(N)}$	Approximation of Čech complex
Alpha	$N^{O(\lceil d/2 \rceil)}$	Nerve Theorem
Sparse Čech	$O(N)$	Approximation of Čech complex
Sparse VR	$O(N)$	Approximation of VR complex

Table 3.1: Some simplicial complexes, the worst-case sizes of the complexes as functions of the cardinality N of the vertex set and their theoretical guarantees[74].

The general steps for the computation of a simplicial complex are shown in Figure 3.8 and can be described as follows:

1. A PCD is given and balls around its points are built.
2. An edge linking every couple of points whose balls intersect is added.
3. A filled triangle for every triplette of points whose balls intersect each other is added.
4. n-dimensional polytypes are added generalizing the same logic of points 2 and 3.

In the next sections Čech, Vietoris–Rips and Sparse Čech complexes are described.

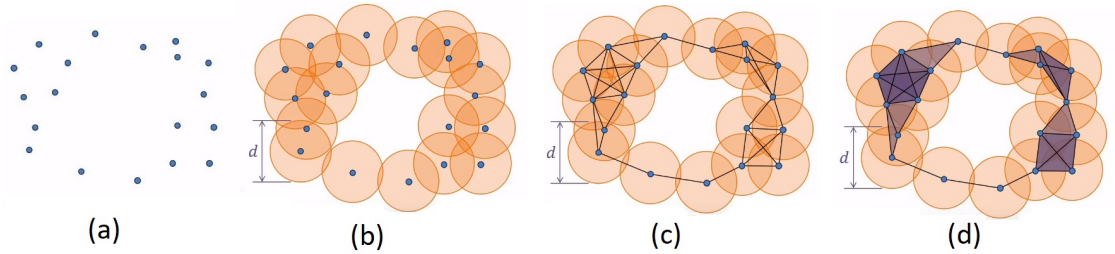


Figure 3.8: In (a) the PCD is shown, in (b) a distance d, called the proximity parameter, is choosen and in (c) the nearby points are connected by edges. In (d) the VR simplicial complex is built. Figure from [125].

3.1.3 Čech Complexes

Definition 3.26. Given a PCD X and a proximity parameter $r > 0$, the *Čech complex* $\check{C}_r(X)$ is build as

$$\check{C}_r(X) := \{[p_1, p_2, ..., p_k] \mid \{p_1, p_2, ..., p_k\} \subset X, \cap_i B(p_i, r) \neq \varnothing\}$$

where $B(p,r)$ is the closed ball of radius r centered at p.

In Figure 3.9 an example is displayed.

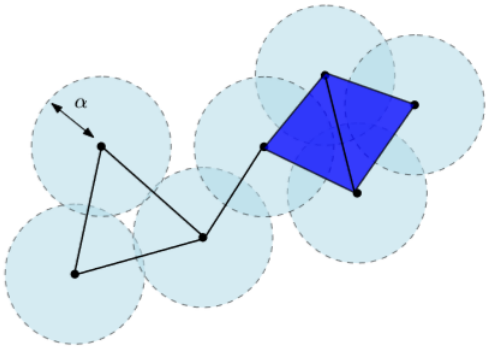


Figure 3.9: The Čech complex $\check{C}_\alpha(X)$ of a finite point cloud in the plane $\mathbb{R}^2$. It's dimension is 2. Figure from [48].

The *nerve* of a covering is a construction of an abstract simplicial complex from a covering of a topological space X.

Definition 3.27. Given a covering $\mathscr{U} = \{U_i\}_{i \in I}$ of a topological space X, the *nerve* of U is the abstract simplicial complex $Nrv(\mathscr{U})$ whose vertices are the U_i's and such that

$$\sigma = [U_{i_0}, ..., U_{i_k}] \in Nrv(\mathscr{U}) \Leftrightarrow \cap_{j=0,...,k} U_{i_j} \neq 0$$

For an example of nerve, see Figure 3.10.

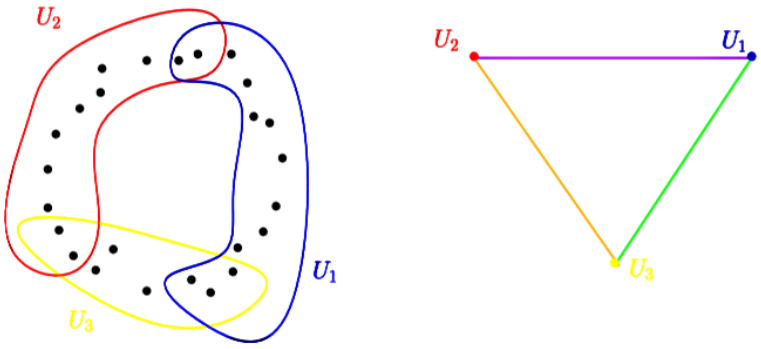


Figure 3.10: The nerve of a cover of a set of points. Figure from [48].

Theorem 3.28 (Nerve Theorem[48])**.** *Given a topological space X and an open cover $\mathscr{U} = \{U_i\}_{i\in I}$ of it such that the intersection of any subset of the U_i's is either empty or contractible, X and the nerve $Nrv(\mathscr{U})$ are homotopy equivalent.*

Definition 3.29. Given two topological spaces X and Y, A function $f : X \to Y$ is *proper* if the preimage of every compact set in Y is compact in X.

Definition 3.30. Given a compact subset of $\mathbb{R}^d$ $K = \{x_0, ..., x_n\}$:

- Given $r \geq 0$, the *r-offset* of K is the union of balls of radius r centered on K.
- Given $r \geq 0$, the *r-sublevel set* of K is the distance function

$$d_K : \mathbb{R}^d \to \mathbb{R},\ d_K(x) := inf_{y\in K}||x-y||$$

- A function $\phi : \mathbb{R}^d \to \mathbb{R}_+$ is *distance-like* if it is proper and $x \to ||x||^2 - \phi^2(x)$ is convex.
- Let ϕ be a distance-like function and let $\phi^r = \phi^{-1}([0,r])$ be the r-sublevel set of ϕ. A point $x \in R^d$ is α-*critical* if $||\nabla_x \phi|| \leq \alpha$.
- For any $0 < \alpha < 1$, the $reach_\alpha$ of ϕ is the maximum r such that $\phi^{-1}((0,r])$ does not contain any α-critical point.

Theorem 3.31 (Reconstruction Theorem[48])**.** *Let ϕ, ψ be two distance-like functions such that $||\phi - \psi||_\infty < \varepsilon$, with $reach_\alpha(\phi) \geq R$ for some positive ε and α. Then, for every $r \in [4\varepsilon/\alpha^2, R - 3\varepsilon]$ and every $\rho \in (0,R)$, the sublevel sets ψ^r and ϕ^ρ are homotopy equivalent when $\varepsilon \leq \frac{R}{5+\frac{4}{\alpha^2}}$.*

The Reconstruction Theorem combined with the Nerve Theorem tell that, for well-chosen values of r and ρ, the ρ-offsets of K are homotopy equivalent to the nerve of the union of balls of radius r centered on K, $\check{C}_r(K)$[48]. This means that this complex gives faithful representation of data.

See Figure 3.11.

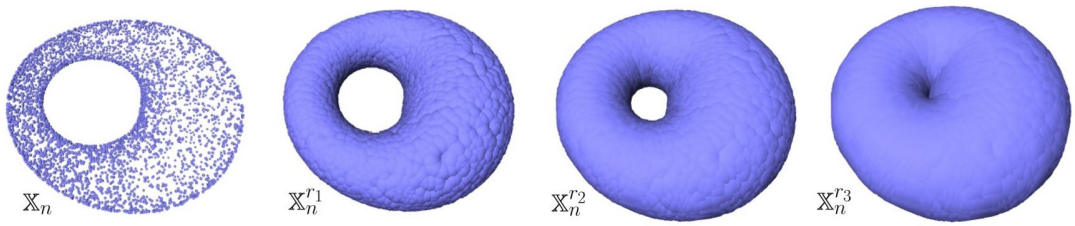


Figure 3.11: The example of a PCD sampled on the surface of a torus in $\mathbb{R}^3$ (top left) and its offsets for different values of r. For r_1 and r_2, the offsets are homotopy equivalent to a torus. Figure from [48].

The Reconstruction Theorem requires the choice of a radius r and a regularity assumption through $reach_\alpha$ that may not be satisfied. To address these issues the PH will be introduced. Before of that, we first introduce other two types of simplicial complexes derived from the Čech one: Vietoris–Rips and Sparse Čech complexes.

3.1.4 Vietoris–Rips Complexes

Definition 3.32. Given a PCD X and an $r > 0$, the *Vietoris–Rips complex* $VR_r(X)$ is

$$VR_r(X) := \{[p_1, p_2, ..., p_k] \mid \{p_1, p_2, ..., p_k\} \subset X, max_{p_i,p_j \in \sigma}(dist(p_i, p_j)) \leq r\}$$

Note that if *dist* is $||\cdot||_\infty$, then $VR_{2r}(X) = \check{C}_r(X)$[39]. In Figure 3.12 an example is displayed, while in Figure 3.13 a comparison between VR and Čech complexes is shown.

Proposition 3.33. If $X \subset \mathbb{R}^d$ then $\check{C}_\alpha(X)$ and $VR_{2\alpha}(X)$ have the same set of vertices and edges[48].

The Čech complex is a subcomplex of the VR complex but it is more computationally expensive because of the higher number of intersections of the balls in the complex[74].

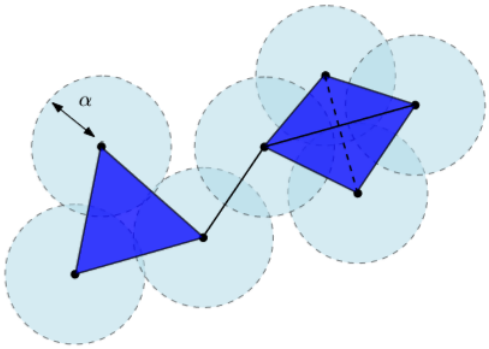


Figure 3.12: The $VR_{2\alpha}$ complex of the PCD in the plane $\mathbb{R}^2$ of Figure 3.9. It's dimension is 3. Figure from [48].

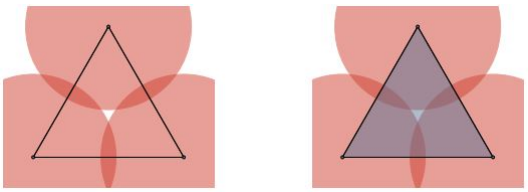


Figure 3.13: Comparison of $\check{C}_r$ (left) and VR_{2r} (right) complexes. Figure from [112].

However, the Nerve Theorem provides a guarantee that the Čech complex is homotopy equivalent to union of the balls in the complex, while VR complex may not be[69] but consider the proposition below[112][48].

Proposition 3.34. For every finite set of points $K \subset \mathbb{R}^d$ and $r \geq 0$,

$$\check{C}_r(K) \subset VR_{2r}(K) \subset \check{C}_{2r}(K)$$

Thus, if $\check{C}_r(K)$ and $\check{C}_{2r}(K)$ approximates the data in a good way, then $VR_{2r}(K)$ do it as well and this estimate can be improved[72].

Proposition 3.35. For every finite set of points $K \subset \mathbb{R}^d$ and $r \geq 0$,

$$VR_{r'}(K) \subset \check{C}_r(K) \; if \; \frac{r}{r'} \geq \sqrt{\frac{2d}{d+1}}$$

.

On one hand, VR is ideally suited to communication networks, since the entire complex is determined by pairwise structures. On the other, it doesn't necessarily capture the topology of the union of cover discs[72]. In Figure 3.14 a PCD example for which the VR fails to capture the Čech complex is given.

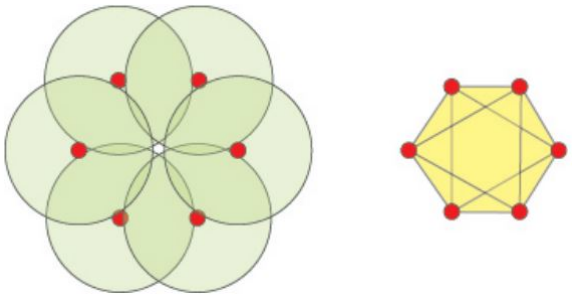


Figure 3.14: The Čech complex is homotopy equivalent to a circle. The VR one however is homeomorphic to S^2. Figure from [72].

Čech complex is difficult to calculate, but it is quite small and accurate. However, VR complex is easy to calculate, but is usually very big[112] and less accurate.

Both complexes can produce a simplicial complex of dimension greater than the considered space. To build simplicial complexes with few simplices that approximate the homology of a space and are easy computable, we present another alternative: the Sparse Čech complex.

3.1.5 Sparse Čech Complexes

For large proximity parameters few points of the PCD is needed to provide a good approximation, see Figure 3.15. The idea behind Sparse Čech complex is to consider less points as the chosen radius increases. As we will see, it approximates the PH of the Čech complex but have fewer simplices than it.

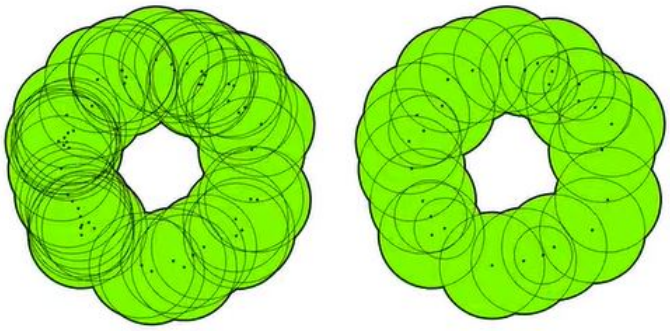


Figure 3.15: The two complexes are similar but the right one actually uses fewer points. Figure from [40].

Before defining it, we need some new definitions[39].

Definition 3.36. Given $P = \{p_1, ..., p_n\}$, $P_i = \{p_1, ..., p_i\}$ is the *i-th prefix*. P is ordered according to a *greedy permutation* if and only if $\forall i \in \{2, ..., n\}$, $dist(p_i, P_{i-1}) = max_{p \in P} dist(p, P_{i-1})$. For each point p_i, the value $\lambda_i = dist(p_i, P_{i-1})$ is the *insertion radius*. By convention, $\lambda_1 = \infty$.

To build a sparse version of the Čech complex we define new radius and $r-$balls concepts. Then, we apply them to the definition 3.26 so that as the radius increases, only a sparse subset of points keeps contributing to the offsets. The concepts is that some balls will be completely covered by their neighbors ones[39].

Definition 3.37. Given $P = \{p_1, ..., p_n\}$ ordered by a greedy permutation with insertion radii $\lambda_1, ..., \lambda_n$ and a constant $\varepsilon < 1$, the *radius* of p_i at scale α is

$$r_i(\alpha) := \begin{cases} \alpha, & \text{if } \alpha \leq \dfrac{\lambda_i(\varepsilon+1)}{\varepsilon} \\ \dfrac{\lambda_i(\varepsilon+1)}{\varepsilon}, & \text{otherwise} \end{cases}$$

Definition 3.38. Given a point p_i, with insertion radio λ_i, radius r_i and a constant $\varepsilon < 1$, the $\alpha-$*ball* of p_i is

$$b_i(\alpha) := \begin{cases} B(p_i, r_i(\alpha)), & \text{if } \alpha \leq \dfrac{\lambda_i(\varepsilon+1)^2}{\varepsilon} \\ \varnothing, & \text{otherwise} \end{cases}$$

Definition 3.39. The *Sparse Čech complex* is defined as

$$Q_\alpha(X) := \{[p_1, p_2, ..., p_k] \mid \{p_1, p_2, ..., p_k\} \subset X, \cap_i b_i(\alpha) \neq \varnothing\}$$

Proposition 3.40. $b_i(\alpha) = \varnothing$ unless λ_i is large enough compared to α, so fewer balls are considered as the scale increases. Note that the number of vertices does not change.

The justification based on PH results for the use of this complex can be found in Theorem 4.13. However, the "vertex removals" mentioned in proposition 3.40 were proved to be implementable as a sequence of elementary edge collapses.

The key result is that a collapse, in general, reduces a simplicial complex to a homotopy-equivalent subcomplex. The resulting subcomplex is homotopy equivalent to the original one if the *link condition* holds[59].

Definition 3.41. Let K be an abstract simplicial complex. Suppose $\tau, \sigma \in K$ such that the following two conditions are satisfied:

- $\tau \subset \sigma$, in particular $\dim(\tau) < \dim(\sigma)$,
- τ is a facet.

A *simplicial collapse* of K is the removal of all simplices γ such that $\tau \subseteq \gamma \subseteq \sigma$, where τ is a free face. If $dim(\tau) = dim(\sigma) - 1$, then this is called an *elementary collapse*.

The collapse is performed by identifying pairwise vertices as shown in Figure 3.16.

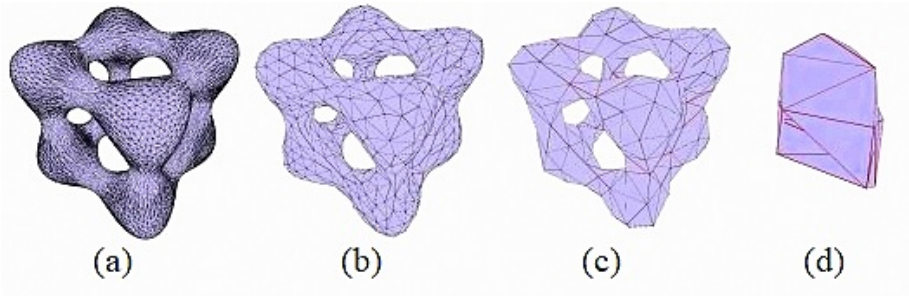


Figure 3.16: A simplicial complex simplified with different numbers of collapses. In (a) the VR complex with $\sim 70 \cdot 10^6$ simpleces is shown. Its resulting complexes after 6000, 6700 and 6787 are shown in (b), (c) and (d) respectively. In the last case the number of simplices is ~ 100. Figure from [114].

The *link* of a simplex σ in a complex K is $Lk(\sigma) = \{\tau \setminus \sigma | \tau \in K, \sigma \subseteq \tau\}$. An edge $\{u,v\} \in K$ satisfies the link condition if and only if $Lk\{u,v\} = Lk\{u\} \cap Lk\{v\}$. See Figure 3.17.

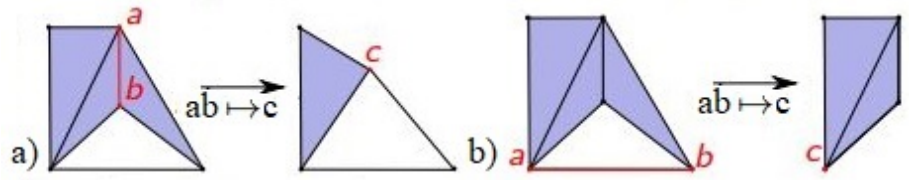


Figure 3.17: An example of collapse satisfying the link condition (left) an one of collapse not satisfying it (rigth). Figure rearranged from [114].

Proposition 3.42. [39] If (P,d) is a finite subset of a convex metric space and $\{S^\alpha\}$ is its corresponding sparse filtration, then the last vertex p_n has a neighbor p_i such that the edge $\{p_n, p_i\} \in S^\alpha$ satisfies the link condition, where $\alpha = \lambda_n(1+\varepsilon)2/\varepsilon$ and λ_n is the insertion radius of p_n.

3.1.6 Filtrations

Definition 3.43. A filtration is a family of subsets $\{X_a | a \in A\}$ indexed by a totally ordered set A such that $X_a \subset X_b$ for $a \leq b$.

Note that given a simplicial complex, with the increase of the distance d, it becomes more complicated. In particular, the simplicial complex generated by the distance $d_1 < d_2$ is included in the one generated by the distance d_2. The sequence of simplicial complexes with its inclusion maps is a filtration called *filtered simplicial complex*. See Figure 3.18.

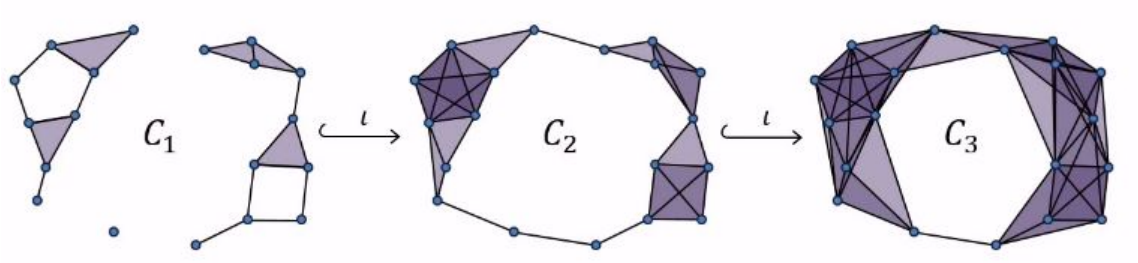


Figure 3.18: A filtration segment rapresentation on the simplicial complex of Figure 3.8. Image from [125].

In particular, the Čech filtration, VR filtration and Sparse Čech filtration are respectively defined as $\{\check{C}_\alpha\}_{\alpha>0}$, $\{VR_\alpha\}_{\alpha>0}$ and $\{\cup_{\delta\leq\alpha} Q_\delta\}_{\alpha>0}$ equiped with their inclusion maps. This last definition is motivated by the fact that $\{Q_\alpha\}_{\alpha>0}$ is not a filtration[39].

Simplicial complexes provides a topologically faithful summary of the data, but they are not well-suited for further processing. We need easier computable topological descriptors, in particular numerical ones. This issue will be managed by considering the homology.

3.2 Homology

The original motivation for defining homology groups was the observation that two shapes can be topologically distinguished by examining their holes. For instance, a circle is not topologically equivalent to a disk because the circle has a hole. However, it's not trivial to find holes with common techniques due to the fact that they are not "present".

The fundamental group $\pi_1(X)$ is especially useful when studying loops and homotopies of loops of low dimension spaces. However, the higher-dimensional homotopy groups are extremely difficult to compute[81].

A more computable alternative to homotopy groups are homology groups $H_n(X)$.

An important property of these algebraic structures is that they are robust as they are homotopy invariant. An example from [81] is now proposed to show what the idea behind homology is.

3.2.1 Simplicial Homology Example

Consider the graph in Figure 3.19 (a). It consists of two vertices joined by four edges.

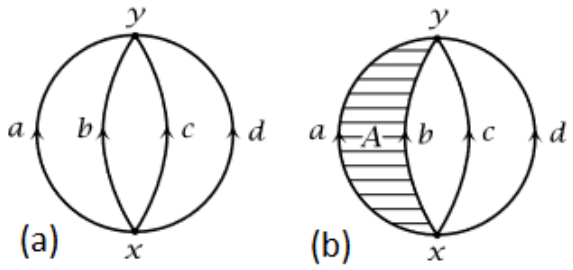


Figure 3.19: Homology example.

Consider the *chain of edges* $a-b+c-d$ travelling forward edge a, then backward along b and so on. Some of these chains can be decomposed into cycles in several ways, for example $[a-c]+[b-d]=[a-d]+[b-c]$, but we don't want to distinguish between these decompositions.

A geometric cycle is characterized by the fact that it enters and leaves each vertex the same number of times. Generally, let C_1 be the free abelian group with basis a,b,c,d that are 1-dimensional chains. Let C_0 be the free abelian group with basis x,y that are linear combinations of vertices or 0-dimensional chains.

We can define a homomorphism $\partial : C_1 \to C_0$ by mapping each element of the basis a,b,c,d into $y-x$. We get $\partial(ka+lb+mc+nd)=(k+l+m+n)y-(k+l+m+n)x$ and the cycles are the kernel of ∂ whose basis consists of $a-b$, $b-c$ and $c-d$. Thus

every cycle in Figure 3.19 (a) is a unique linear combination of these three cycles. We infer that the graph has three holes.

Let glue a 2-cell A along the cycle $a-b$, see Figure 3.19 (b). If A is oriented clockwise, we can regard its boundary as the cycle $a-b$. This cycle can be now contracted to a point, so it no longer encloses a hole. This suggests that we have quotientated the group of cycles by factoring the subgroup generated by $a-b$.

We can define the homomorphisms $C_2 \xrightarrow{\partial_2} C_1 \xrightarrow{\partial_1} C_0$ where C_2 is the infinite cyclic group generated by A and $\partial_2(A) = a-b$. The quotient group we are interested in is the 1-dimensional cycles modulo those that are boundaries, the multiples of $a-b$, that is the homology group $H_1(X_2) = Ker(\partial_1)/Im(\partial_2)$. Here $H_1(X_2)$ has two generators, $b-c$ and $c-d$, so we have reduced the number of holes to two.

3.2.2 Simplicial Homology

All polyhedra can be decomposed into simplices so there is no loss of generality in focusing on simplices. Homology detects k-dimensional holes in a simplicial complex X imposing an algebraic structure on it. For each $k \geq 0$, an abstract vector space C_k (k-chains) is built with basis consisting of the set of k-simplices in K, so that the dimension of C_k equals the number of k-simplices[76].

To define a basis, we have to choose an ordering of all the vertices and give to each simplex the induced corresponding orientation. Let $\sigma = [v_0, ..., v_k]$ be an oriented k-simplex, viewed as a basis element of C_k. The boundary operator $\partial_k : C_k \to C_{k-1}$ is the homomorphism $\partial_k(\sigma) = \sum_{i=0}^{k}(-1)^i(v_0, \ldots, \widehat{v_i}, \ldots, v_k)$, where the oriented simplex $[v_0, \ldots, \widehat{v_i}, \ldots, v_k]$ is the i-th face of σ, obtained by deleting its i-th vertex. The signs are inserted to take orientations into account, so that all the faces of a simplex are coherently oriented, as indicated in the Figure 3.20.

Observation 3.44. It holds that $\partial^2 = 0$, meaning that a boundary has no boundary[76]. For example, $\partial^2([v_0, v_1, v_2]) = \partial([v_1, v_2]) - \partial([v_0, v_2]) + \partial([v_0, v_1]) = [v_2] - [v_1] - [v_2] + [v_0] + [v_1] - [v_0] = 0$.

In C_k, elements of the subgroup $Z_k = \ker \partial_k$ are referred to as cycles, and the subgroup $B_k = \operatorname{im} \partial_{k+1}$ consists of boundaries. From the observation 3.44 it follows that B_k is a

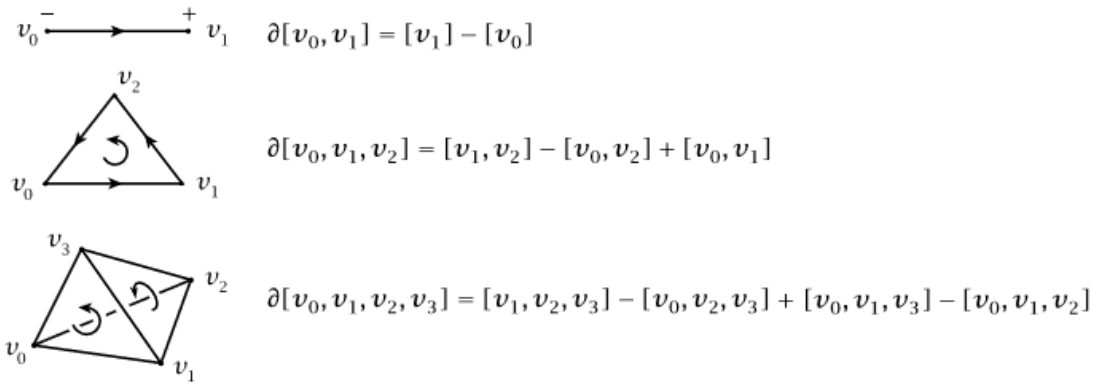


Figure 3.20: k-simplices boundariers. Image from [81].

subspace of Z_k. The goal of homology is to discard cycles that are also boundaries, so we'll quotientiate Z_k using the following equivalence relation.

Definition 3.45. Two cycles $z_1, z_2 \in Z_k$ are *homologous* if they differ by a boundary, that is $z_1 \sim z_2 \Leftrightarrow z_1 - z_2 \in B_k$.

Example 3.46. Consider the example in Figure 3.21 provided by [76]. The blue chain $b = [v_0, v_1] + [v_1, v_2] + [v_2, v_3] + [v_3, v_4] + [v_4, v_0]$ and the red chain $r = [v_1, v_2] + [v_2, v_3] + [v_3, v_4] + [v4, v1]$ are cycles because $\partial(b) = \partial(r) = 0$. These cycles are homologous because their difference is a the green boundary, $g = b - r = [v_0, v_1] + [v_4, v_0] - [v_4, v_1] = [v_1, v_4] - [v_0, v_4] + [v_0, v_1] = \partial[v_0, v_1, v_4]$.

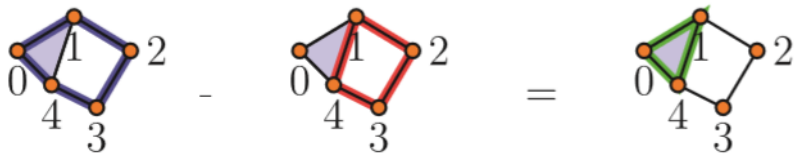


Figure 3.21: The blue and the red cycles are homologous because their difference is the boundary of the green triangle. Image from [76].

Definition 3.47. The *k-th homology group* H_k of a simplicial complex S is the quotient abelian group $H_k(S) = Z_k / B_k$.

The elements of $H_k(K)$ are the equivalence classes of homologous cycles. $H_k(S)$ is non-zero exactly when there are k-cycles on S which are not boundaries meaning that there are k-dimensional holes in the complex. The rank of $H_k(S)$ is the k-th Betti number of S, $\beta_k = \text{rank}(H_k(S)) = dim(Z_k) - dim(B_k)$.

β_0, β_1 and β_2 count respectively the number of connected components, the number of holes and the number of voids in X.

Simplicial homology groups and Betti numbers are topological invariants: if K, K_0 are two simplicial complexes whose geometric realizations are homotopy equivalent, then their homology groups are isomorphic and they have the same Betti numbers, see Figure 3.22.

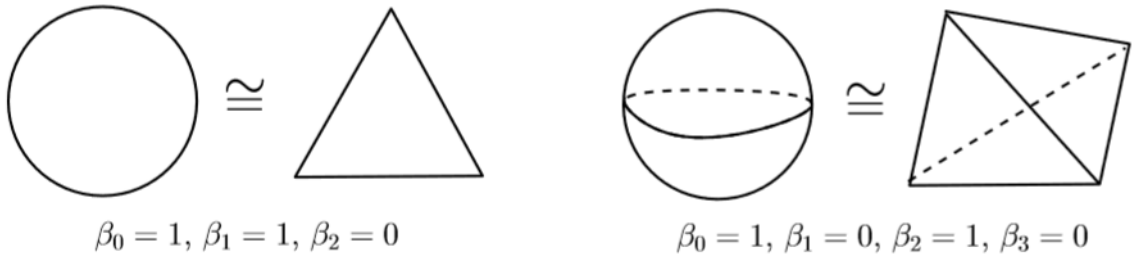


Figure 3.22: The Betti numbers of the circle (left) and the 2-dimensional sphere (right). Image from [48].

Another example is presented below.

Example 3.48. Let S be the triangle without its interior as a simplicial complex. Thus S has three vertices v_0, v_1 and v_2 and three edges. To compute the homology groups of S, we start by describing the chain groups C_k.
C_0 is isomorphic to $\mathbb{Z}^3$ with basis v_0, v_1, v_2. C_1 is isomorphic to $\mathbb{Z}^3$ with a basis given by the oriented 1-simplices $[v0, v1]$, $[v0, v2]$, and $[v1, v2]$. The chain groups in other dimensions are zero.
The boundary homomorphism $\partial : C_1 \to C_0$ is given by $\partial[v_0, v_1] = v_1 - v_0$, $\partial[v_0, v_2] = v_2 - v_0$ and $\partial[v_1, v_2] = v_2 - v_1$. B_0 is generated by the three elements on the right of these equations, so $H_0(S) = Z_0/B_0$ is isomorphic to $\mathbb{Z}$ with a basis given by the image of the 0-cycle v_0 (all three vertices become equal in the quotient group). So S is connected.
The group of 1-cycles is $ker(\partial)$ which is isomorphic to $\mathbb{Z}$, with a basis given, for example, by $[v_0, v_1] - [v_0, v_2] + [v_1, v_2]$. Since $C_2 = 0$, the group of 1-boundaries is zero, and so the homology group $H_1(S)$ is isomorphic to $\mathbb{Z}/0 \sim \mathbb{Z}$. So, the triangle has one 1-dimensional hole.

We can finally introduce the PH, but before of that a little overview on singular homology is provided.

Singular Homology

Simplicial homology requires the space to be triangulated but not every space can be, and even if it is, it is not necessarily true that the triangulation is unique.

Singular homology assigns homology groups to every topological space encoding invariants of the space in an analogous way as simplicial homology assigns homology groups to simplicial complexes[74].

Rather than decompose the space into simplices, it considers the collection of all possible continuous maps of simplices into X. These maps generate extremely large chain groups but the quotients, called *singular homology groups*, turn out to be generally smaller. However in simplicial homology, unlike singular one, doing calculations is quite straightforward and we'll no further discuss singular homology.

4. Persistent Homology and Stability

Our goal is to recover the properties of the underlying space of data robustly to small perturbations. To do that, PH will be now introduced. Consider, for example, the simplicial complex in Figure 4.1. How to choose the right proximity parameter d? If d is too small (a), we might see multiple distinct components and small holes that can be a result of the noise. If d is to big (c), we get a giant simplex with a trivial topology. In the Figure (b) the distance d reveals a single hole, but we need to have some highlight to understand if it's a true feature of the data[125].

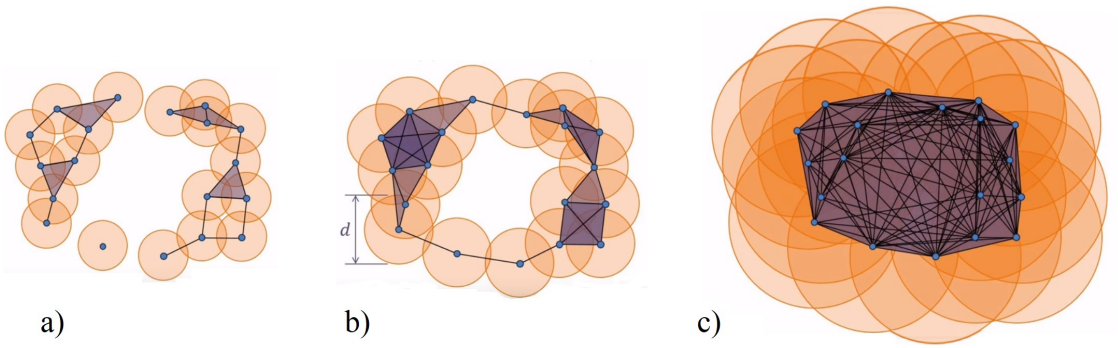


Figure 4.1: Simplicial complex examples built on different distance d values. Image from [125].

In order to do that, we can consider all the distances $d > 0$. Note that each hole appears at a particular value d_1, and disappears at another value d_2.

Given a parameterized family of spaces, those topological features which persist over a significant parameter range are to be considered as true features with short-lived characteristics due to noise[70]. We can represent the persistence of the hole with a segment $[d_1, d_2)$, see Figure 4.2.

Figure 4.2: Persistence of an hole appering for distance d_1 and disappering for distance d_2 as a bar. Image from [125].

A collection of such bars is a *persistence barcode*, see Figure 4.3.

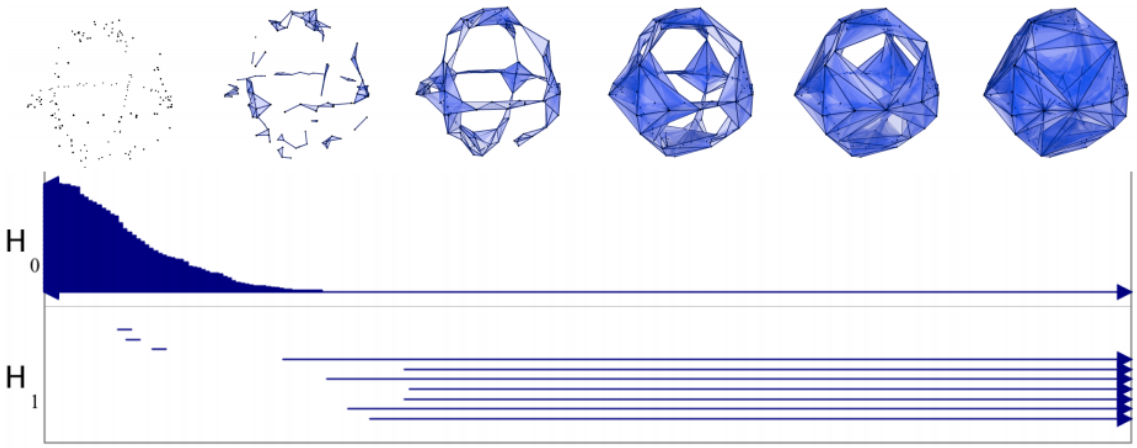


Figure 4.3: A filtration and its barcode. Image from [39].

An alternative graphical way to represent barcodes is the *persistence diagram*, in which an interval $[i, j)$ is represented by the point (i, j), see Figure 4.4.

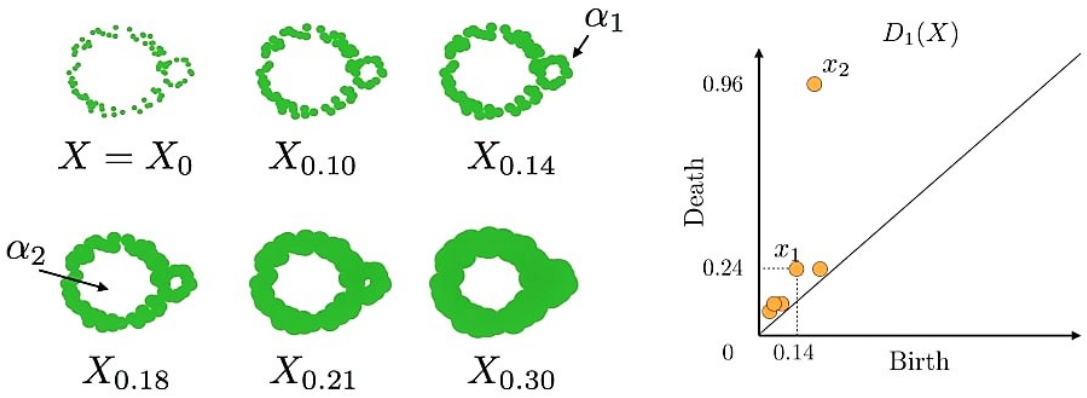


Figure 4.4: On the right the union X_r of r-balls at points sampled from annuli with noise. On the left, the persistence diagram in which x_1 represents the ring α_1, which born at $r = 0.14$ and dies at $r = 0.24$. The noisy rings are plotted as the points close to the diagonal. Image from [65].

With PH we study the homology of a filtered simplicial complex as a single algebraic entity. Its features can be then studied using its barcode and this is justified formally by the Structure Theorem.

4.1 From PH to Barcodes

As we explained, given a PCD and derived a simplicial complex X_r with proximity parameter r, $H(X_r)$ is a vector space that is the quotient of the k-cycles modulo those that are boundaries. As r increases, the union of disks grows and the resulting inclusions induce maps between the homology groups.

Example 4.1. Assume that $VR = (VR_i)_1^N$ is a sequence of VR complexes associated to a fixed PCD for an increasing sequence of parameter values $(\varepsilon_i)_1^N$. The inclusions $VR_{\varepsilon_1} \hookrightarrow VR_{\varepsilon_2} \hookrightarrow ... \hookrightarrow VR_{\varepsilon_N}$ holds. Instead of examining the homology of the individual terms VR_i, we examine the homology of the iterated inclusions $H(VR_i) \to H(VR_j)$ for all $i < j$.

Definition 4.2. When $0 \leq i \leq j \leq n$, the inclusion $x_i^j : K_i \hookrightarrow K_j$ induces a homomorphism $H_p(x_i^j) : H_p(K_i) \to H_p(K_j)$ on the simplicial homology groups for each dimension p. The p^{th} *persistent homology groups* are the images of these homomorphisms, and the p^{th} *persistent Betti numbers* $\beta_p^{i,j}$ are the ranks of those groups

Observation 4.3. PH groups explain why VR complexes are an acceptable approximation to Čech complexes[72], as mentioned in proposition 3.34. For any $r > 0$, there is a chain of inclusion maps $VR_r(K) \hookrightarrow \check{C}_r(K) \hookrightarrow VR_{2r}(K)$. So, although no single Rips complex is an especially faithful approximation to a single Čech one, pairs of Rips complexes 'squeeze' the appropriate Čech complex into a manageable hole.

Definition 4.4. Let X be a topological space and f a real function on X. A *homological critical value* of f is a real number a for which there exists an integer k such that for all sufficiently small $\varepsilon > 0$ the map $H_k\left(f^{-1}(-\infty, a-\varepsilon]\right) \to H_k\left(f^{-1}(-\infty, a+\varepsilon]\right)$ induced by inclusion is not an isomorphism.

Definition 4.5. A *persistence module* M is a vector space $\{M_a\}_{a\in\mathbb{R}}$ with the linear maps $M(a \leq b) : M_a \to M_b, \forall a \leq b$ such that $M(a \leq a)$ is the identity map and $\forall a \leq b \leq c, M(b \leq c) \circ M(a \leq b) = M(a \leq c)$.

Observation 4.6. Given any real-valued function $f : S \to \mathbb{R}$ on a topological space S, we can define the associated persistence module, $M(f)$, where $M(f)(a) = H(f^{-1}((-\infty, a]))$ and $M(f)(a \leq b)$ is induced by inclusion.

Definition 4.7. The *k-th persistence module* h_k is the family of vector spaces $H_k(X_*)$ together with homomorphisms $H_k(x_*^{*'})$.

Observation 4.8. The k-th persistence module can be given the structure of a graded module over the polynomial ring $R[x]$[121]:

$$h_k = \bigoplus_{i=0}^{\infty} H_k(X_i) \cdot F$$

The Structure Theorem states the existence of a simple description of persistent modules as a set of intervals, the barcode.

The β found in Theorem 2.34 is the Betti number of the module. When R is $\mathbb{Z}$, the theorem describes the structure of finitely generated abelian groups. Over a field, the torsion portion disappears and therefore, the module H_k is a vector space fully described by the rank β. The graded ideals of $F[x]$ are of the form $x^n \cdot F[x]$, where multiplication by x corresponds to moving forward one step in the persistence module.

Theorem 4.9. *There is a classification of persistence modules over a field F indexed by $\mathbb{N}$:*

$$U \simeq \bigoplus_i x^{t_i} \cdot F[x] \oplus \left(\bigoplus_j x^{r_j} \cdot (F[x]/(x^{s_j} \cdot F[x])) \right)$$

.

Intuitively, the free parts on the right side correspond to the homology generators that appear at filtration level t_i and never disappear, while the torsion parts correspond to those that appear at filtration level r_j and last for s_j steps.

Definition 4.10. A *barcode* is a finite set of intervals that are bounded below.

Definition 4.11. Given a filter $\sigma = \{\sigma_i\}$, a *persistence barcode* is a set of intervals such that if a simplex σ_i creates a homology class at time s which is destroyed at time t, $0 \leq s < t \leq \infty$, then the interval $[s,t)$ is added to the corresponding persistence barcode. If a simplex σ_j creates a homology class at time s which survives along the process, the interval $[s,\infty)$ is added to the persistence barcode.

We summarize the relation between PH groups and persistence barcodes in the following theorem from [70].

Theorem 4.12. *Given a persistent homology group $H_*(x_i^j)$, its rank is equal to the number of intervals in its barcode spanning the parameter interval $[i, j)$.*

So, persistence modules capture the information contained in the homomorphisms and are classifiable in terms of a compact combinatorial object called a barcode[43]. In Figure 4.5 a more complete example of barcode is provided and in Figure 4.6 the steps from PCD to barcodes are schematized.

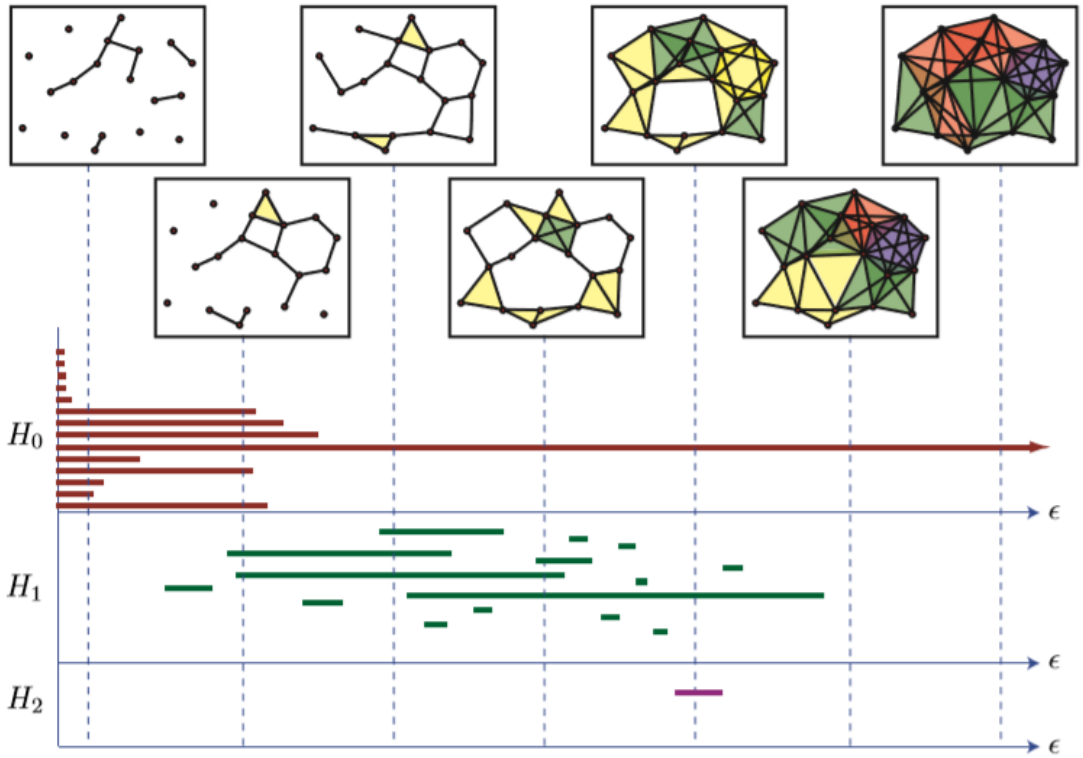


Figure 4.5: [bottom] An example of the barcodes for $H_*(\mathbf{C})$. [top] The rank of $H_k(\mathbf{C}_{\varepsilon_i})$ equals the number of intervals in the barcode for $H_k(\mathbf{C})$ intersecting the dashed line $\varepsilon = \varepsilon_i$. Image from [70].

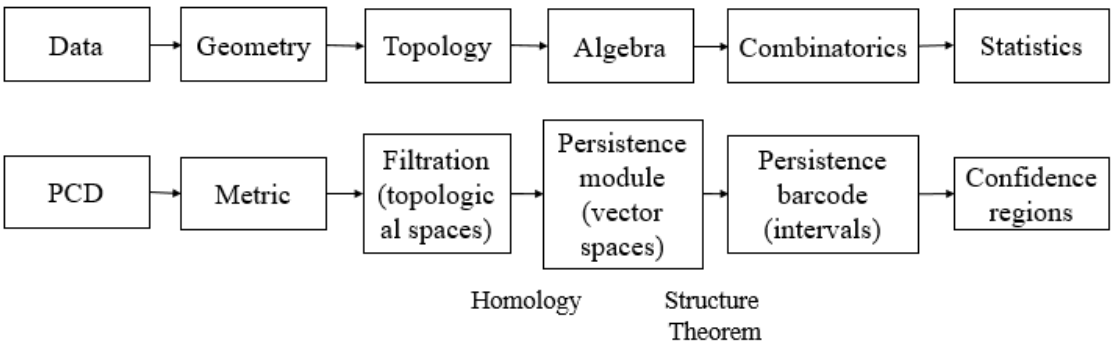


Figure 4.6: TDA pipeline. Image from [14].

To justify the use of the sparse filtration introduced in the previous sections, we report the following theorem by [39].

Theorem 4.13. *The persistence barcode of the sparse nerve filtration* $\{S_\alpha\}_{\alpha\geq 0}$ *is a* $(1+\varepsilon)$*-approximation to the persistence barcode of the original offsets* $\{P_\alpha\}_{\alpha\geq 0}$.

While barcodes provide an intuitive representation of persistence, persistence diagrams are widely used to compute interesting measurements although they provide the same information.

A *multiset* is a generalization of the classical concept of set that allows repeated components.

Definition 4.14. A *persistence diagram* is the union of a finite multiset of points in $\mathbb{R}^2$ with the multiset of points on the diagonal $\{(x,y)\in\mathbb{R}^2 : x=y\}$ where each point on the diagonal has infinite multiplicity.

Definition 4.15. Let D be a persistence diagram. For $x=(b,d)\in D$, let $\ell=d-b$ denote the *persistence* of x. If $D=\{x_j\}$, let $\mathrm{Pers}_k(D)=\sum_j \ell_j^k$ denote the *degree-k total persistence* of D.

If we map the intervals $[i,j)$ of Definition 4.11 into points (i,j) of the persistence diagram we get that each point corresponds to a feature and its importance is proportional to the absolute difference between the two coordinates of the point.

The points on the plane diagonal are included with infinite multiplicity because this allows to give every persistence diagram the same cardinality and so compare them by studying bijections between their elements[74]. This is used in the study of the stability of persistent diagrams.

Some examples of applications are now provided to help the reader getting an idea of how barcodes can be interpreted.

Analysis of Fullerene Structure[123]. In Figure 4.7, the PH analysis of icosahedron and fullerene C_{70} are shown.

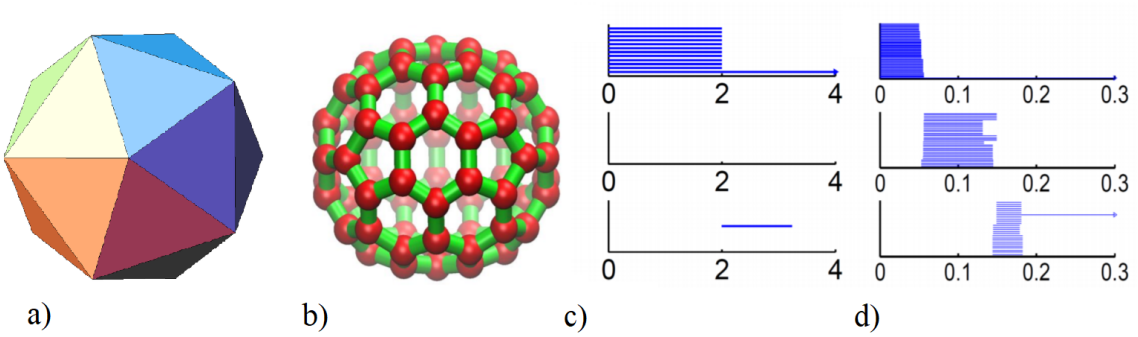


Figure 4.7: Persistent homology analysis of the icosahedron (a) and fullerene C_{70} (b) are shown respectively in (c) and (d) where there are three panels corresponding to β_0, β_1 and β_2 bars, respectively. Images from [123].

Note that for the icosahedron:

- β_0: Originally 12 bars coexist, indicating 12 isolated vertices. Then, 11 of them disappear simultaneously with only one survived. These vertices connect with each other at $\varepsilon = 2\mathring{A}$, i.e., the designed bond length. The positions where the bars terminate are exactly the corresponding bond lengths.
- β_1: As no one-dimensional circle has ever formed, no circle is generated.
- β_2: There is a single bar, which represents a two-dimensional void enclosed by the surface of the icosahedron.

In regarding of the fullerene C_{70} barcodes:

- β_0: There are 70 initial bars and 6 distinct groups of bars due to the presence of 6 types of bond lengths in the C_{70} structure.
- β_1: There is a total of 36 bars corresponding to 12 pentagon rings and 25 hexagon rings. It appears that one ring is not accounted because any individual ring can be represented as the linear combination of all other rings. Note that there are 6 types of rings.
- β_2: 25 hexagon rings further evolve to two-dimensional holes, which are represented by 25 bars. The central void structure is captured by the persisting β_2 bar.

Topological Fingerprints of Proteins[123]. Two most important protein structural components, namely, alpha helices and beta sheets, are analyzed to reveal their unique topological features, which can be recognized as their *topological fingerprints*. In Figure

4.8 an alpha helix an its coarse-grained (CG) model are topologically compared.

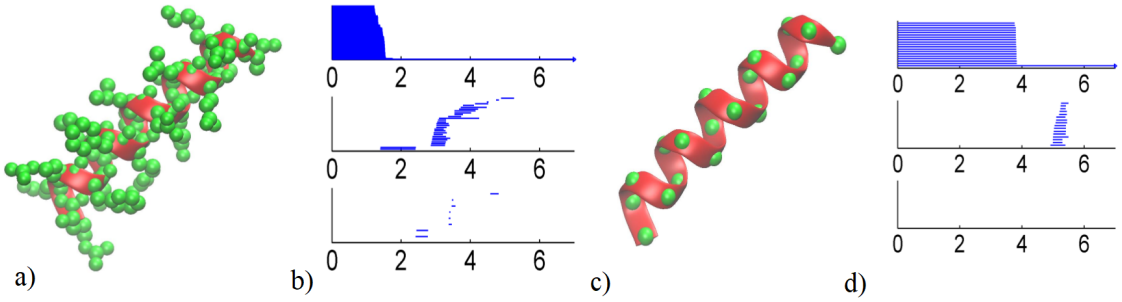


Figure 4.8: Persistent homology analysis of the alpha helix structure (a) and CG model (c) are shown respectively in (b) and (d) where there are three panels corresponding to β_0, β_1 and β_2 bars, respectively. Atoms are demonstrated in green color and the helix structure of the main chain backbone is represent by the cartoon shape in red. Images from [123].

The characteristic distance c is the relative influence domain of the atoms of biomolecules. Usually, for the CG model (in witch each amino acid is represented by its C_α atom) the optimized cut off distance is about $7 - 8\mathring{A}$. Optimal characteristic distances, however, can be revealed from PH analysis.

As seen, the β_0 bars can be very useful to reveal the bond length information. In Figure a-4.8, we can observe that the helix alpha structure backbone has a loop-type structure, but the corresponding barcode does not clearly demonstrate these patterns due to the fact that there are two many atoms around the main chain. To extract more geometric and topological details of the helix structure, we use the CG model.

As there are 19 residues in the alpha helix structure, only 19 atoms are used in the CG model and the corresponding barcode is dramatically simplified. In Figure d-4.8, it is seen that there are 19 β_0 bars and the bar length is around $3.8\mathring{A}$, which is the average length between two atoms. Additionally there are 16 β_1 bars with similar birth time and persist length. To reveal the topological meaning of these bars, we make use of a technique called *slicing*.

Basically, we slice a piece of 4 atoms from the backbone and study its persistent homology behavior. Then, one more atom is added at a time. The results are shown in Figure 4.9.

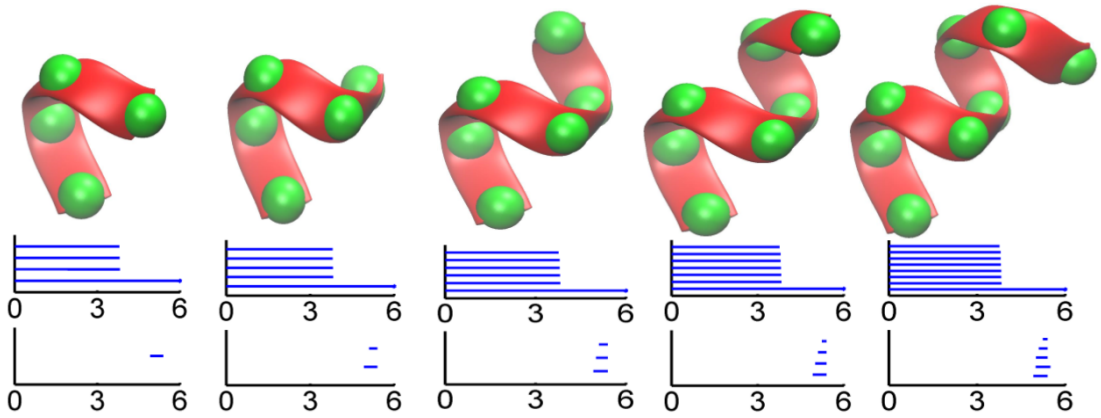


Figure 4.9: Method of slicing for the analysis of alpha helix topological fingerprints. In the coarse-grain representation, each residue is represented by a C_α atom. In an alpha helix. Images from [123].

It can be seen that each four atoms in the alpha helix form a one-dimensional loop, corresponding to a β_1 bar. By adding more atoms, more loops are created and more β_1 bars are obtained. Finally, 19 residues in the alpha helix produce exactly 16 loops as seen in Figure 4.8. Each loop is contributed from 4 C_α.

4.2 Stability

In this paragraph we'll show that, under some assumptions, the persistence diagram is stable with respect to small perturbations: little changes in the data imply only small changes in the diagram.

To compare persistence diagrams we'll endow the space of persistence diagrams with *bottleneck distance*, however consider that there are many variants of the stability results for persistence diagrams, as we may define different distances between persistence diagrams.

Definition 4.16. A *multi-bijection* is a bijective map between two multi-sets counted with their multiplicity.

Definition 4.17. The Bottleneck distance between two persistence diagrams D_1 and D_2 is

$$d_B(D_1, D_2) = inf_\gamma sup_{x \in D_1} ||x - \gamma(x)||_\infty$$

where γ ranges over all multi-bijections from D_1 to D_2.

Note that d_B satisfies all axioms of a metric and thus deserves to be called a distance. See Figure 4.10 for an example. For this definitions to make sense, we add infinitely many copies of every point on the horizontal axis to the diagrams on the diagonal so they guarantee that there are bijections between the multisets.

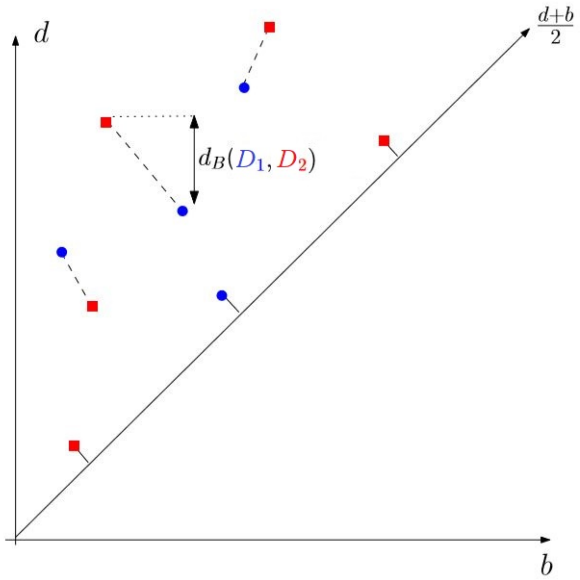


Figure 4.10: The Bottleneck distance between a blue and a red diagram. Image from [48].

However, the bottleneck metric is completely determined by the largest distance among the pairs and do not take into account the closeness of the remaining pairs of points[48]. A variant, to overcome this issue, is the *Wasserstein distance*.

Definition 4.18. Given two persistence diagrams X and Y and $p \in [1,\infty]$, the *p-th Wasserstein distance* between X and Y is

$$W_p[d](X,Y) := \begin{cases} inf_{\Psi:X\to Y}(\sum_{x\in X} d[x,\Psi(x)]^p)^{\frac{1}{p}}, \ for\ p \in [1,\infty) \\ inf_{\Psi:X\to Y}(sup_{x\in X} d[x,\Psi(x)]), \ for\ p = \infty \end{cases}$$

where d is a metric on $\mathbb{R}^2$ and Ψ ranges over all bijections from X to Y.

Note that $d_B = W_\infty[L^\infty]$. The Wasserstein distance is defined by finding the perfect pairing that minimizes the sum, rather than the supremum, of the pairwise distances[12]. It is more sensitive than bottleneck distance to details in the diagrams but requires additional properties to be stable. The bottleneck distance is cruder but leads to a more general result, so from now on we will focus on it.

Definition 4.19. A function is *tame* if it has only finitely many homological critical values, and all sublevel sets have finite rank homology groups.

In words, f is tame if for all but finitely many $a \in R$, the associated persistence module $M(f)$ is constant and finite dimensional on some open interval containing a[22].

Theorem 4.20 (Stability Theorem for Tame Functions)**.** *Let X be a triangulable topological space and $f, g : X \to R$ two tame functions. For each dimension p,*

$$d_B(D_p(f), D_p(g)) \leq ||f - g||_\infty$$

See Figure 4.11.

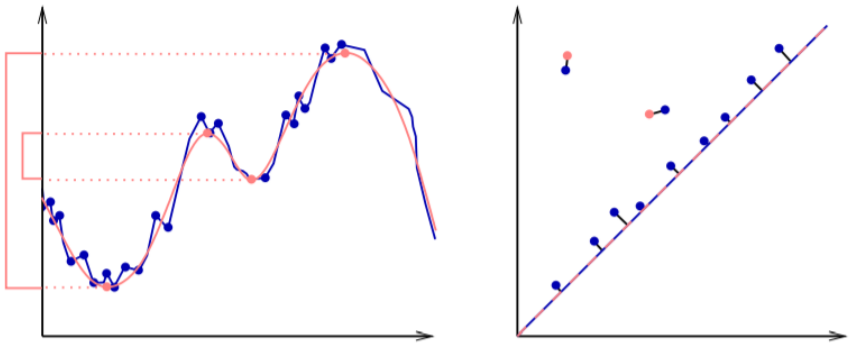


Figure 4.11: Left: two close functions, one with many and the other with just four critical values. Right: the persistence diagrams of the two functions, and the bijection between them. Image from [52].

The assumptions required for this result are mild and are satisfied by Morse functions on compact manifolds, piecewise linear functions on simplicial complexes, and more.

Theorem 4.21 (Stability Theorem for Filtrations)**.** *Let K be a simplicial complex and $f, g : K \to R$ two monotonic functions. For each dimension p, it holds that*

$$d_B(D_p(f), D_p(g)) \leq ||f - g||_\infty$$

The bottleneck distance is based on a bijection between the points and is therefore always at least the Hausdorff distance between the two diagrams [52]. So, we can get a lower bound of $||f - g||_\infty$ using the d_H that is generally easier to compute and to approximate.

Proposition 4.22. Let X and Y be finite subsets in a metric space (M, d_M), then

$$d_B(D(X), D(Y)) \leq d_H(X, Y)$$

.

This gives a geometric intuition of the stability of persistence diagrams. Assume that X is the true location of points and Y is a data obtained from skewed measurement with $\varepsilon = d_H(X,Y)$. If there is a point $(b,d) \in D(Y)$, then we can find at least one generator in X which is born in $(b-\varepsilon, b+\varepsilon)$ and dies in $(d-\varepsilon, d+\varepsilon)$. See Figure 4.12.

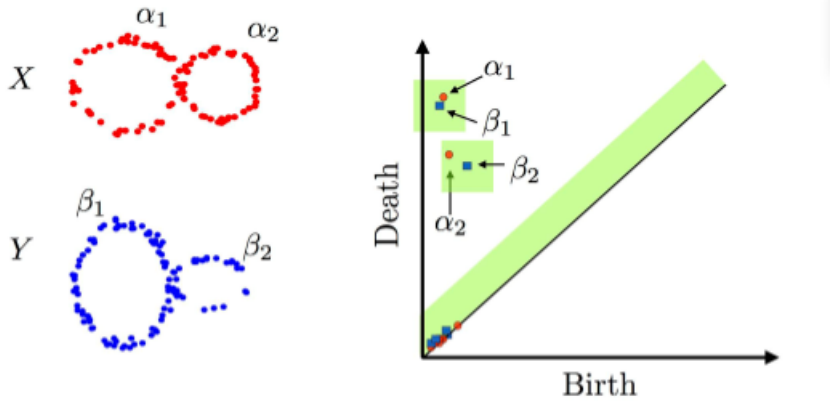


Figure 4.12: Two data X and Y (left) and their persistence diagrams (right). The green region is an ε – neighborhood of $D_q(Y)$. Image from [65].

More generally, we can obtain the following result[46].

Theorem 4.23. *Let X and Y be two compact metric spaces and let $Filt(X)$ and $Filt(Y)$ be the Čech (or VR) filtrations built on top them. Then*

$$d_B(D(Filt(X)), D(Filt(Y))) \leq 2 d_{GH}(X,Y)$$

Moreover, if X and Y are embedded in the same space then

$$d_B(D(Filt(X)), D(Filt(Y))) \leq 2 d_H(X,Y)$$

We need methods for quantitatively assessing the quality of our results. In the next chapter, we will discuss some statistical approaches to this problem.

5. Statistical Discussion

Some of the main goals of a statistical approach are to provide confidence regions for topological features and select relevant scales at which the topological phenomenon should be considered.

We'll consider data as generated from an unknown distribution. The topological features inferred by TDA methods will be seen as estimators of the topological quantities of the true object[48]. We now report three of the main methods used for the statistical analysis of PH results[74]:

- Compare the simplicial complexes built on empirical data to random simplicial complexes used as null models;
- Study the properties of a metric space whose points are persistence diagrams;
- Map the space of persistence diagrams to Banach spaces amenable to statistical analysis and machine-learning techniques. Such methods include *persistence landscapes*.

We'll now briefly describe and discuss these methods.

5.1 Random Simplicial Complexes

Non-local properties of networks are not expected to be closely reproduced by random graphs with only local constraints[13]. For example, the global properties of human brain differ drastically from the ones of a random graph whose degree distributions, degree correlations and clustering are the same of the brain ones.

Following the paper [106], we introduce some definitions.

Definition 5.1. The *degree* d_i of a node v_i is the number of facets incident on it.

Definition 5.2. The *size* s_i of a facet σ_i is the number of nodes it contains.

This local information can be summarized by $d = (d_1, ..., d_n)$ and $s = (s_1, ..., s_f)$, where n is the number of nodes and f is the number of facets.

Definition 5.3. The *simplicial configuration model* (SCM) is the uniform distribution over all labeled simplicial complexes with degree sequence d and facet size sequence s.

SCM allows describing arbitrary complexes in order to obtain a generic null model.

Observation 5.4. Let $\Omega(d,s)$ be the set of all labeled simplicial complexes with joint sequences (d,s). Then if SCM has sequences (d,s), it places a probability

$$P(K;d,s) = \begin{cases} 1/|\Omega(d,s)| \text{ on } K \\ 0 \text{ on otherwise} \end{cases}$$

Let's now switch to the equivalent graphical representation of simplicial complexes. We denote by F the facets set, and by $V \cup F$ the complete node set. Facets are replaced by nodes, and an edge connects facet $\sigma_i \in F$ to node $v_j \in V$ if and only if σ_i is incident to v_j in K, see Figure 5.1.

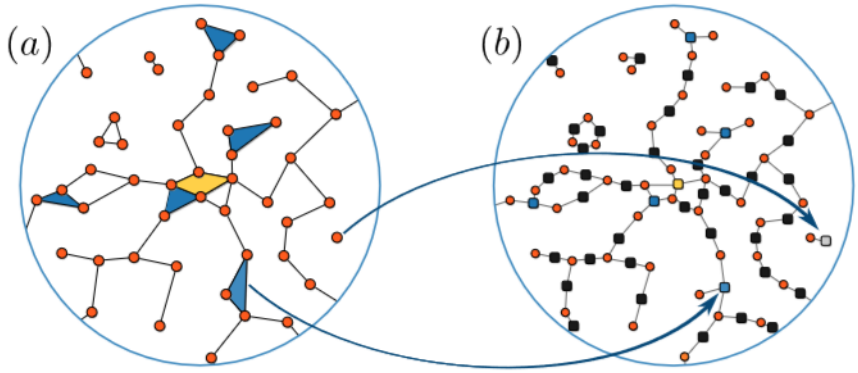


Figure 5.1: (a) Simplicial complex K and (b) its graphical representation. Image from [106].

Sampling from the SCM of parameters (d,s) is not equivalent to uniformly sampling from all bipartite graphs with these degree sequences because the mapping is not bijective.

Definition 5.5. A bipartite graph with joint degree sequences (d,s) is *sequence preserving* if its equivalent simplicial complex has facet size sequence s and generalized degree sequence d.

See Figure 5.2.

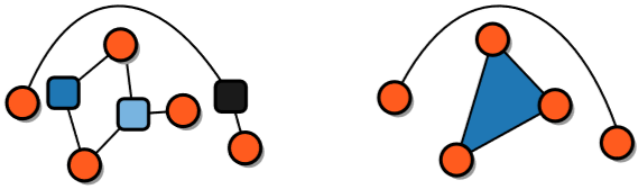


Figure 5.2: (a) Example of non-degree-preserving bipartite graphs. Image from [106].

A *Markov chain* is a stochastic model describing a sequence of possible events in which the probability of each event depends only on the state attained in the previous event[66]. Given a $k > 0$, when the distribution of the state X_{t+1} of the chain is the same as the distribution of X_t for all $t > k$ we talk of *equilibrium distribution*.

Markov Chain-based Monte Carlo (MCMC) methods are a class of algorithms for sampling from probability distributions based on the construction of a Markov chain having as equilibrium distribution the desired distribution. In [106], the MCMC sampling strategy is used. The idea is to build a random chain of sequence preserving bipartite graphs, to sample from it at regular intervals, and to treat the samples as if they had been drawn i.i.d..

Since every instance of the SCM has the same fixed local structure but is maximally random, we expect significant differences between the Betti numbers of an organized simplicial complex and the bulk of the distribution of β in the corresponding randomized ensembles.

Three datasets were analysed an it was found that the distribution of β_0 and β_1 for the SCM associated was essentially random in one case. That is, the overwhelming majority of simplicial complexes with the same sequences have similar β_0 and β_1. See Figure 5.3.

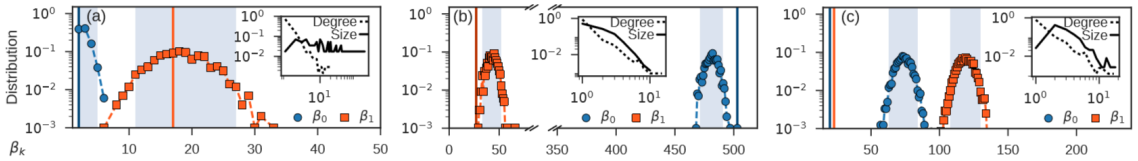


Figure 5.3: The Betti numbers of these real systems appear as vertical lines. The distributions of Betti numbers for the equivalent SCM with solid symbols (computed from 1000 instances of the model) are shown. The shaded regions contain 95% of the samples. The distributions on the left are associated with random features, while those in the middle and on the right differ from the distributions of the random counterparts. Image from [106].

In contrast, the β of the other two datasets were highly different. The researcher could conclude that the shape of the first dataset was completely determined by its local structure, while large–scale organizational principles influence the structure of the other ones.

In the discussion of the last two methods, we will use the *bootstrap technique*. Because of this, now a brief explanation on it is provided.

5.2 The Bootstrap

The bootstrap is a general method that can be used for computing confidence intervals[12].

Definition 5.6. A $(1-\alpha)$-confidence interval for a parameter θ is an interval $[a,b]$ such that the probability $P(\theta \in [a,b])$ is at least $1-\alpha$.

Given a measure space (X,Ω,P), let $X_1,\ldots,X_n$ be i.i.d. random variables taking values on it. If we want to estimate parameter θ related to the distribution P of the observation we can use the statistic $\widehat{\theta} = g(X_1,\ldots,X_n)$, which is some function of the data. For example, θ and $\widehat{\theta}$ could be the population mean and the sample mean, respectively.

Given the cumulative distribution F of $\hat{\theta}-\theta$, the quantiles $F^{-1}(1-\alpha/2)$ and $F^{-1}(\alpha/2)$ can be computed. Calculating $a = \hat{\theta} - F^{-1}(1-\alpha/2)$ and $b = \hat{\theta} - F^{-1}(\alpha/2)$ we can obtain a $1-\alpha$ confidence interval for θ as

$$\mathbb{P}(\theta \in [a,b]) = \mathbb{P}\left(F^{-1}\left(\frac{\alpha}{2}\right) \leq \hat{\theta}-\theta \leq F^{-1}\left(1-\frac{\alpha}{2}\right)\right) = 1-\alpha$$

However, F depends on the unknown distribution P. So, we approximate it with the empirical measure P_n that puts mass $1/n$ at each X_i in the sample.

Let's sample $X_1^*,\ldots,X_n^*$ from $X_1,\ldots,X_n$ with replacement. Then, we can estimate the distribution $F(r)$ with the distribution

$$\widehat{F}(r) = P_n\left(\hat{\theta}^* - \hat{\theta} \leq r\right), \text{ where } \hat{\theta}^* = g\left(X_1^*,\ldots,X_n^*\right)$$

The distribution $\widehat{F}$ is still not analytically computable, but can be approximated by simulation. For large B, obtain B different values of $\hat{\theta}^*$ and approximate $\widehat{F}(r)$ with

$$\widetilde{F}(r) = \frac{1}{B}\sum_{i=1}^{B} I\left(\hat{\theta}_i^* - \hat{\theta} \leq r\right).$$

Since the quantiles of $\tilde{F}$ approximate the quantiles of F, we define the estimated confidence interval as

$$C_n = \left[\hat{\theta} - \widetilde{F}_n^{-1}(1-\alpha/2), \hat{\theta} - \widetilde{F}_n^{-1}(\alpha/2)\right]$$

Summarizing, with the bootstrap we:

1. create a random sample with replacement from the original sample with sample size as the original sample,
2. calculate the sample statistic
3. repeat steps 1 and 2 B times to obtain the bootstrap distribution
4. use this bootstrap distribution to calculate confidence intervals.

See Figure 5.4. We will get accurate estimates only if the sample size is sufficiently large. Formally, one has to show that

$$\sup_r |\widetilde{F}(r) - F(r)| \xrightarrow{P} 0$$

which implies that

$$\liminf_{n\to\infty} \mathbb{P}(\theta \in C_n) \geq 1-\alpha$$

where C_n is the confidence interval.

An *empirical process* is a stochastic process based on a random sample.

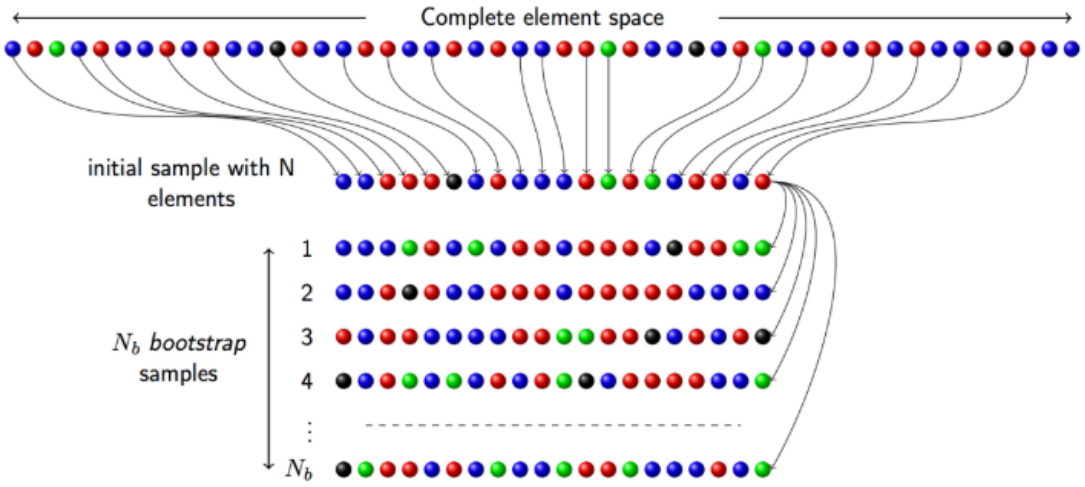


Figure 5.4: The bootstrap procedure. Image from [115].

Definition 5.7. Given a measure space (X,Ω,P), let $X_1,...,X_n$ be i.i.d. random variables taking values in it. For a measurable function $f:\mathbb{X}\to\mathbb{R}$, we denote $Pf=\int fdP$ and $P_nf=\int fdP_n=n^{-1}\sum_{i=1}^n f(X_i)$.

Proposition 5.8. By the law of large numbers P_nf converges almost surely to Pf.

The *bootstrap empirical process*, can be used to find a confidence band for a function $h(t)$ that is two functions $a(t)$ and $b(t)$ such that

$$\mathbb{P}(h(t)\in[a(t),b(t)])\geq 1-\alpha\ \forall t$$

Definition 5.9. Given a class $\mathscr{F}$ of measurable functions, we define the empirical process $\mathbb{G}_n$ indexed by $\mathscr{F}$ as $\{\mathbb{G}_nf\}_{f\in\mathscr{F}}=\{\sqrt{n}(P_nf-Pf)\}_{f\in\mathscr{F}}$.

Definition 5.10. $\ell^\infty(\mathscr{F})$ is the collection of all bounded functions $f:\mathbb{X}\to\mathbb{R}$.

Definition 5.11. *A* class $\mathscr{F}$ of measurable functions $f:\mathbb{X}\to\mathbb{R}$ is *P-Donsker* if a process $\{\mathbb{G}_nf\}_{f\in\mathscr{F}}$ converges in distribution to a limit process in the space $\ell^\infty(\mathscr{F})$.

Proposition 5.12. The limit process to which $\{\mathbb{G}_nf\}_{f\in\mathscr{F}}$ converges is a Gaussian process G with zero mean and covariance function $Pfg-PfPg$.

Definition 5.13. Let $P_n^*f=n^{-1}\sum_{i=1}^n f(X_i^*)$ where $\{X_1^*,\ldots,X_n^*\}$ is a bootstrap sample from P_n, that is the measure that puts mass $1/n$ on each element of the sample

$\{X_1,\dots,X_n\}$. The bootstrap empirical process $\mathbb{G}_n^*$ indexed by $\mathscr{F}$ is defined as

$$\{\mathbb{G}_n^* f\}_{f\in\mathscr{F}} = \left\{\sqrt{n}\,(P_n^* f - P_n f)\right\}_{f\in\mathscr{F}}$$

Theorem 5.14. *[12] $\mathscr{F}$ is P -Donsker if and only if $\mathbb{G}_n^*$ converges in distribution to* G *in* $\ell^\infty(\mathscr{F})$.

Suppose we are interested in constructing a confidence band of level $1-\alpha$ for $\{Pf\}_{f\in\mathscr{F}}$, where $\mathscr{F}$ is P -Donsker. Let $\hat{\theta} = \sup_{f\in\mathscr{F}} |\mathbb{G}_n f|$. We proceed as follows:

- we sample $X_1^*,\dots,X_n^*$ from P_n and compute $\hat{\theta}^* = \sup_{f\in\mathscr{F}} |\mathbb{G}_n^* f|$
- we repeat the previous step B times to obtain $\hat{\theta}_1^*,\dots,\hat{\theta}_B^*$
- we compute $q_\alpha = \inf\left\{q : \frac{1}{B}\sum_{j=1}^{B} I\left(\hat{\theta}_j^* \geq q\right) \leq \alpha\right\}$
- for $f\in\mathscr{F}$, we define the confidence band $C_n(f) = \left[P_n f - \frac{q_n}{\sqrt{n}}, P_n f + \frac{q_g}{\sqrt{n}}\right]$

A consequence of Theorem 1.4 is that, for large n and B, the interval $[0, q_\alpha]$ has coverage $1-\alpha$ for $\hat{\theta}$ and the band $C_n(f)_{f\in\mathscr{F}}$ has coverage $1-\alpha$ for $\{Pf\}_{f\in\mathscr{F}}$.

5.3 Distance Approach

Using the Stability Theorem 4.23, we can define confidence sets to separate topological signal from topological noise. Given a persistence diagram $\mathscr{D}$ with an estimator $\widehat{\mathscr{D}}$, we look for some value η_α such that

$$P\left(\mathrm{d_B}(\widehat{\mathscr{D}},\mathscr{D}) \geq \eta_\alpha\right) \leq \alpha \; for \; \alpha\in(0,1)$$

The confidence set related will be

$$\left\{\mathscr{D} : d_B(\widehat{\mathscr{D}},\mathscr{D}) \leq \eta_\alpha\right\}$$

We can visualize it by adding a box of side length $2\eta_\alpha$ centered at each point on the persistence diagram. Given a point p of the persistence diagram the corresponding box is defined as

$$\{q\in\mathbb{R}^2 : d_\infty(p,q) \leq \eta_\alpha\}$$

If this box intersects the diagonal, p is considered indistinguishable from noise[12]. We can also visualize the confidence set by adding a band of width $\sqrt{2}\eta_\alpha$ around the diagonal.

The points in the band are considered as noise. See Figure 5.5.

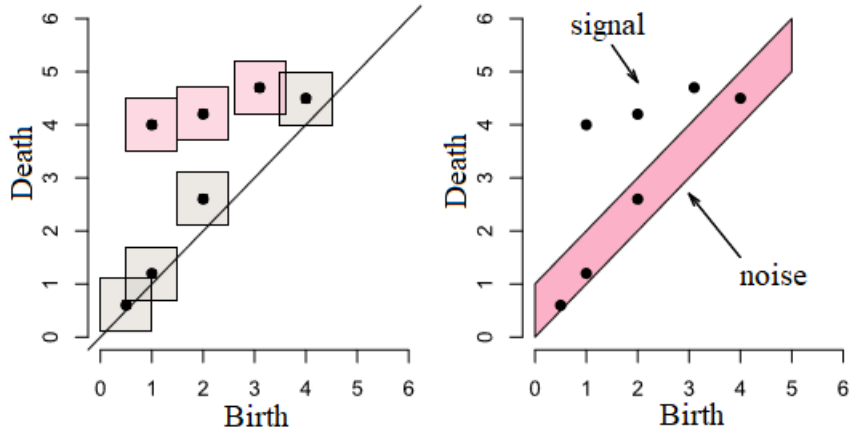


Figure 5.5: Persistence diagram and its confidence region. On the left, the confidence boxes. On the right, the corresponding band of confidence. Image from [12].

Observation 5.15. This trivial separation between signal and noise is not the only way to quantify the uncertainty in the persistence diagram. Indeed, some points near the diagonal may represent interesting structures. One can imagine endowing each point with a different confidence set or assigning it a specific p-value as in [51].

Several methods have been proposed to estimate η_α.

5.3.1 Subsampling Method

Let (M,ρ) be a metric space. Given $X_1,...,X_n$ in M drawn i.i.d. from some unknown measure μ whose support is a compact set X_μ, an *estimator* $\widehat{X}$ of X_μ is a function of $X_1,...,X_n$ that takes values in the set of compact metric spaces and that is measurable for the Borel algebra induced by d_{GH}.

Definition 5.16. Given $a,b>0$, a measure μ satisfies the (a,b)-*standard assumption* if for any $x\in X_\mu$ and $r>0$, $\mu(B(x,r))\geqslant \min\left(ar^b,1\right)$.

According to the Theorem 4.23, we can state the proposition below.

Proposition 5.17. [46]

$$\forall\varepsilon>0,\ \mathbb{P}\left(\mathrm{d_b}\left(\mathrm{D}\left(\mathrm{Filt}\left(X_\mu\right)\right),\mathrm{D}(\mathrm{Filt}(\widehat{\mathrm{X}}))\right)>\varepsilon\right)\leqslant\mathbb{P}\left(\mathrm{d_{GH}}\left(X_\mu,\hat{X}\right)>2\varepsilon\right)$$

where the probability corresponds to the product measure $\mu^{\otimes n}$.

Using this proposition, we then derive the following result for persistence diagram estimation.

Theorem 5.18. *[12] If the probability measure μ on M satisfies the (a,b)-standard assumption*

$$\forall \varepsilon > 0, \mathbb{P}\left(\mathrm{d_b}\left(\mathrm{D}\left(\mathrm{Filt}\left(\mathbb{X}_\mu\right)\right), \mathrm{D}\left(\mathrm{Filt}\left(\widehat{\mathrm{X}}_n\right)\right)\right) > \varepsilon\right) \leqslant \min\left(\frac{2^b}{a\varepsilon^b e^{\left(na\varepsilon^b\right)}}, 1\right)$$

This theorem can be used to find confidence sets for persistence diagrams. However, they will depend on a and b which may be unknown.

Theorem 5.19. *[12]*

- *Let X_b be a subsample of size b drawn from the sample X_n, where $b = o(n/logn)$.*
- *Let $q_b(1-\alpha)$ be the quantile of of the distribution of $d_H(X_b, X_n)$.*
- *Take $\hat{\eta}_\alpha := 2\hat{q}_b(1-\alpha)$ where $\hat{q}_b$ is an estimation $q_b(1-\alpha)$ using a standard Monte Carlo procedure.*

Under an (a,b)-standard assumption, and for n large enough we can infer

$$P\left(\mathrm{d_b}(\mathrm{D}(\mathrm{Filt}(K))), \mathrm{D}\left(\mathrm{Filt}\left(\mathbb{X}_n\right)\right)\right) > \hat{\eta}_\alpha) \leq P\left(d_H\left(K, \mathbb{X}_n\right) > \hat{\eta}_\alpha\right) \leq \alpha + O\left(\frac{b}{n}\right)^{1/4}$$

An alternative strategy is the *bootstrap method*.

5.3.2 Bootstrap Method

Definition 5.20. A *smooth* function is a function that has derivatives of all orders everywhere in its domain.

Definition 5.21.

- Let $X_1, \ldots, X_n$ be a sample from the distribution P, supported on a smooth manifold $X \subset \mathbb{R}^D$.

- Let $K : \mathbb{R} \to \mathbb{R}$ be an integrable function satisfying $\int K(u)du = 1$ and such that $K(u)$ is non-negative for all u.
- Let $p_h(x) = \int_{\mathbf{X}} \frac{1}{h^D} K\left(\frac{\|x-u\|}{h}\right) dP(u)$.

p_h is a probability distribution called *kernel density*. The function K is called *kernel* and the parameter $h > 0$ is its *bandwidth*.

Definition 5.22. The standard estimator for p_h is the *kernel density estimator*

$$\hat{p}_h(x) = \frac{1}{n}\sum_{i=1}^{n} \frac{1}{h^D} K\left(\frac{\|x - X_i\|}{h}\right)$$

Note that if X_i are fixed, then $\hat{p}_h$ is a probability distribution.

- Given a sample $X_1, \ldots, X_n$ the first step in the bootstrap approach is to compute $\hat{p}_h$.
- Then we sample $X_1^*, \ldots, X_n^*$ from $X_1, \ldots, X_n$ with replacement.
- We can now compute $\theta^* = \sqrt{n}\left\|\hat{p}_h^*(x) - \hat{p}_h(x)\right\|_\infty$, where $\hat{p}_h^*$ is the density estimator computed using $X_1^*, \ldots, X_n^*$.
- Repeat the previous step B times we obtain $\theta_1^*, \ldots, \theta_B^*$.
- We can compute $q_\alpha = \inf\left\{q : \frac{1}{B}\sum_{j=1}^{B} I\left(\theta_j^* \geq q\right) \leq \alpha\right\}$.

Under suitable regularity conditions on the kernel K for which F is Donsker, it holds that

$$\limsup_{n\to\infty} \mathbb{P}\left(\sqrt{n}\,\|\hat{p}_h - p_h\|_\infty > q_\alpha\right) \leq \alpha.$$

We conclude that the $(1-\alpha)$ confidence band for $\mathbb{E}[\hat{p}_h]$ is

$$\left[\hat{p}_h - \frac{q_\alpha}{\sqrt{n}}, \hat{p}_h + \frac{q_\alpha}{\sqrt{n}}\right]$$

5.4 Persistence Landscapes

The two standard topological summaries of data are the barcode and the persistence diagram. However, their spaces lack geometric properties that would make it easily to define basic concepts such as mean, median, and so on. We will define a new closely-related summary, the *persistence landscape*, and then compare it to the two previous summaries. The basic idea is to convert the barcode into a function.

Definition 5.23. Let M be a persistence module. For $a \leq b$, the corresponding *Betti number* of M is given by $\beta^{a,b} = dim(im(M(a \leq b)))$.

Definition 5.24. The *rank function* is the function $\lambda : \mathbb{R}^2 \to \mathbb{R}$ given by

$$\lambda(b,d) = \begin{cases} \beta^{b,d} & \text{if } b \leq d \\ 0 & \text{otherwise} \end{cases}$$

Now let us change coordinates considering $m = \frac{b+d}{2}$ and $h = \frac{d-b}{2}$.

Definition 5.25. The *rescaled rank function* is the function $\lambda : \mathbb{R}^2 \to \mathbb{R}$ given by

$$\lambda(m,h) = \begin{cases} \beta^{m-h,m+h} & \text{if } h \geq 0 \\ 0 & \text{otherwise} \end{cases}$$

Definition 5.26. The *persistence landscape* is a sequence of functions $\lambda_k : \mathbb{R} \to [-\infty, +\infty]$, $\lambda_k(t) := \lambda(t,k)$.

There exist maps in both directions between persistence barcodes (or diagrams) and persistence landscape. To obtain a landscape from a barcode, one replaces every bar of the barcode by a peak, whose height is proportional to the persistence of the bar. In the landscape, we translate all peaks so that they touch the horizontal axis. See Figure 5.6.

Definition 5.27. The persistence landscape corresponding to the barcode B is the set of functions $\{\lambda_k(t) : \mathbb{R} \to \mathbb{R}\}_{k \in \mathbb{N}}$, where $\lambda_k(t)$ is the k^{th} largest value of $\{f_{(a_i,b_i)}(t)\}_{i=1}^{m}$, and $\lambda_k(t) = 0$ whenever $k > m$.

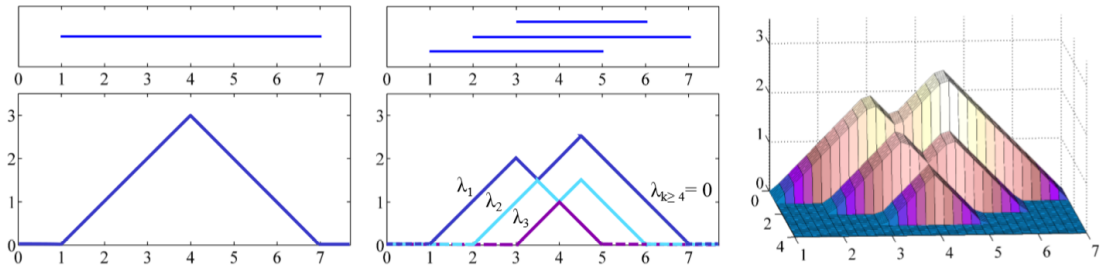


Figure 5.6: On the left, from an interval to the auxiliary function representation. In the middle, from a barcode to a persistence landscape and on the right, the 3D visualization of the persistence landscape. Image from [21].

Definition 5.28. Let M and M' be persistence modules and let λ and λ' be their

corresponding persistence landscapes. For $1 \leq p \leq \infty$, the *p-landscape distance* between M and M' is

$$\Lambda_p\left(M, M'\right) = \left\|\lambda - \lambda'\right\|_p$$

Theorem 5.29 (Landscape Stability Theorem)**.** *Given a real valued function $f : X \to \mathbb{R}$ on a topological space X, let $M(f)$ denote be the corresponding persistence module. Then*

$$\Lambda_\infty(M(f), M(g)) \leq \|f - g\|_\infty$$

Thus the PL is stable with respect to the supremum norm. Note that there are no assumptions on f and g, not even the q-tame condition.

Theorem 5.30. *For persistence diagrams D and D', $\Lambda_\infty\left(D, D'\right) \leq d_B\left(D, D'\right)$.*

So persistence landscape is a stable summary statistic and the landscape distance gives lower bounds for the bottleneck and Wasserstein distances.

The main advantage of the persistence landscapes (PL) over persistence diagrams is that their space is a separable Banach space.

Definition 5.31. A topological space is *separable* if there exists a sequence $\{x_n\}_{n=1}^{\infty}$ of its elements such that every non-empty open subset of it contains at least one element of the sequence.

Definition 5.32. Given a metric space (X, d), a sequence $\{x_n\}$ is *Cauchy*, if $\forall \varepsilon > 0\ \exists N \in \mathbb{N}, N > 0 | \forall m, n \in \mathbb{N}\ with\ m, n > N,\ d(x_m, x_n) < \varepsilon$.

In words, the terms of the sequence get closer in a way that suggests that the sequence ought to have a limit in X. Nonetheless, such a limit does not always exist within X.

Definition 5.33. A *Banach* space is a vector space X over $\mathbb{R}$ or $\mathbb{C}$ equipped with a norm $\|\cdot\|_X$ such that for every Cauchy sequence $\{x_n\}$ in X, $\exists x \in X | \lim_{n\to\infty} \|x_n - x\|_X = 0$.

Sets of persistence diagrams do not have a unique mean, while the space of persistence landscapes does. See Figure 5.7.

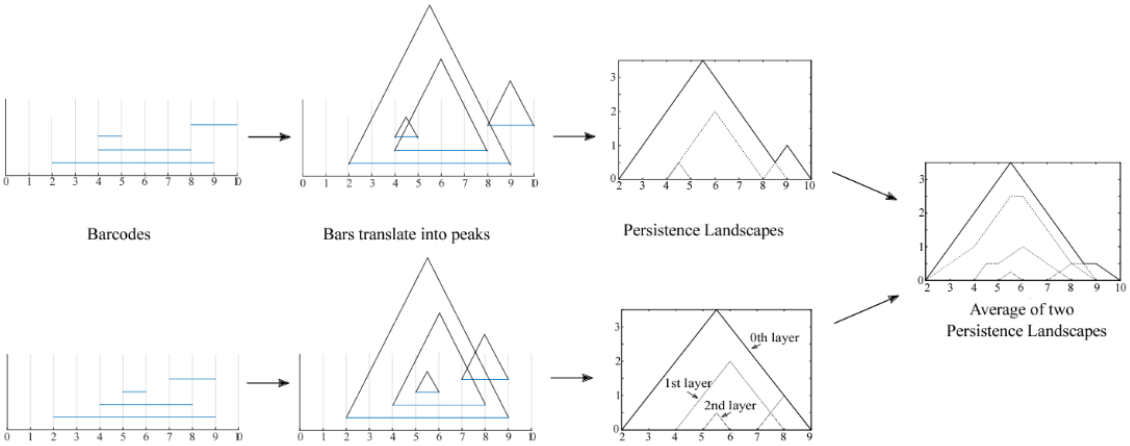


Figure 5.7: We can create an average of two landscapes by taking the mean over the function values in every layer. Image from [80].

Let X be a random variable on a probability space (Ω, F, P). Given $\omega \in \Omega$, $X(\omega)$ is the data and $\Lambda(\omega) = \lambda(X(\omega)) =: \lambda$ is the corresponding persistence landscape.

Definition 5.34. Let $X_1, \ldots, X_n$ be i.i.d. copies of X, and let $\lambda_1, \ldots, \lambda_n$ be the corresponding PLs. The *mean landscape* is given by

$$\overline{\Lambda}_n(\Omega) = \overline{\lambda}_n, \text{ where } \overline{\lambda}_n(k,t) = \frac{1}{n}\sum_{i=1}^{n} \lambda_i(k,t)$$

See Figure 5.8.

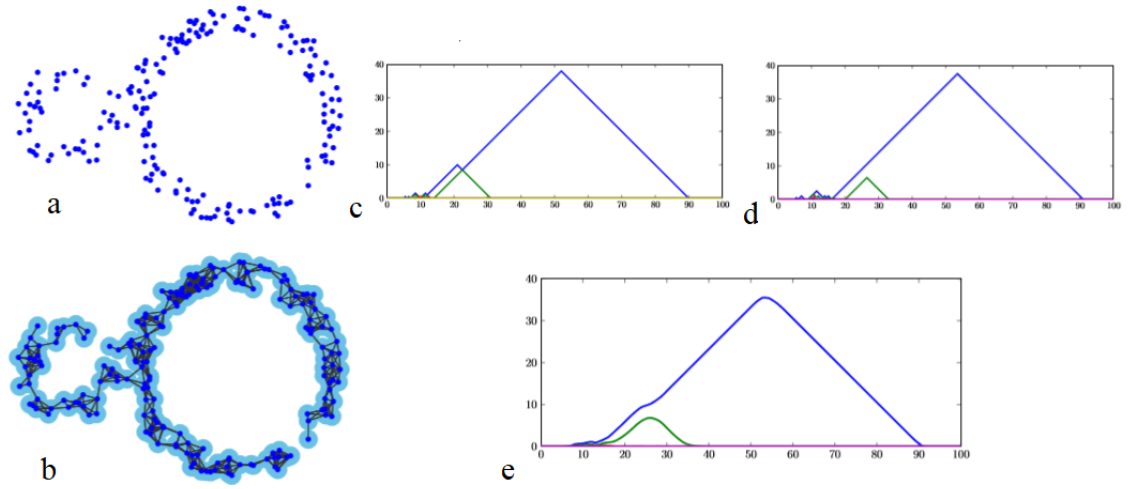


Figure 5.8: 200 points were sampled from a pair of linked annuli and a corresponding union of balls (a) and 1-skeleton of the Čech complex is shown (b). This was repeated 100 times. Two of the one degree persistence landscapes are shown in (c) and (d). Finally, the mean degree one persistence landscape is shown in (e). Image from [22].

To be able to say that the mean landscape converges to the expected persistence landscape we need some notions from probability in Banach spaces[22].

- Let $\mathscr{B}$ be a real separable Banach space with norm $\|\cdot\|$.
- Let $(\Omega, \mathscr{F}, P)$ be a probability space.
- Let $V : (\Omega, \mathscr{F}, P) \to \mathscr{B}$ be a Borel random variable with values in $\mathscr{B}$.
- Let $\mathscr{B}^*$ be the dual space of continuous linear real-valued functions on $\mathscr{B}$.

Proposition 5.35. $\|V\| : \Omega \xrightarrow{V} \mathscr{B} \xrightarrow{\|\cdot\|} \mathbb{R}$ is a random variable.

Proposition 5.36. For $f \in \mathscr{B}^*$, $f(V) : \Omega \xrightarrow{V} \mathscr{B} \xrightarrow{f} \mathbb{R}$ is a random variable.

Definition 5.37. For a random variable $Y : (\Omega, \mathscr{F}, P) \to \mathbb{R}$, the *mean* is

$$E(Y) = \int Y dP = \int_{\Omega} Y(\omega) dP(\omega)$$

Definition 5.38. For a sequence (Y_n) of $\mathscr{B}$-valued random variables, we say that (Y_n) *converges almost surely* to a $\mathscr{B}$-valued random variable Y, if $P(\lim_{n\to\infty} Y_n = Y) = 1$.

Theorem 5.39 (Strong Law of Large Numbers[22])**.** $\left(\frac{1}{n}S_n\right) \to E(V)$ *almost surely* $\Leftrightarrow$ $E\|V\| < \infty$.

Definition 5.40. For a sequence (Y_n) of $\mathscr{B}$-valued random variables, we say that (Y_n) *converges weakly* to a $\mathscr{B}$-valued random variable Y, if $\lim_{n\to\infty} E(\varphi(Y_n)) = E(\varphi(Y))$ for all bounded continuous functions $\varphi : \mathscr{B} \to \mathbb{R}$.

Definition 5.41. A random variable G with values in $\mathscr{B}$ is said to be *Gaussian* if for each $f \in \mathscr{B}^*$, $f(G)$ is a real valued Gaussian random variable with mean zero.

Definition 5.42. The *covariance structure* of a $\mathscr{B}$-valued random variable, V, is given by the expectations $E[(f(V) - E(f(V)))(g(V) - E(g(V)))]$, where $f, g \in B^*$.

Theorem 5.43 (Central Limit Theorem[22])**.** *Let* $\mathscr{B} = L^p(\mathscr{S})$, *with* $2 \leq p < \infty$. *If* $E(V) = 0$ *and* $E\left(\|V\|^2\right) < \infty$ *then* $\frac{1}{\sqrt{n}}S_n$ *converges weakly to a Gaussian random variable* $G(V)$ *with the same covariance structure as* V.

Thanks to these concepts, it's possible to apply the Strong Law of Large Numbers and the Central Limit Theorem for persistence landscapes[22].

Theorem 5.44 (Strong Law of Large Numbers for persistence landscapes)**.** $\overline{\Lambda}^n \to E(\Lambda)$ *almost surely* $\Leftrightarrow E\|\Lambda\| < \infty$.

Theorem 5.45 (Central Limit Theorem for persistence landscapes)**.** *Assume* $p \geq 2$. *If* $E\|\Lambda\| < \infty$ *and* $E\left(\|\Lambda\|^2\right) < \infty$ *then* $\sqrt{n}\left[\overline{\Lambda}^n - E(\Lambda)\right]$ *converges weakly to a Gaussian random variable with the same covariance structure as* Λ.

Example 5.46. In order to perform a hypothesis test, a functional can be applied to each PL, resulting in a single value

$$X = \sum_k \int_{\mathbb{R}} \lambda_k(t)dt$$

It's value is the total area area under all of the persistence landscapes in the k-th homology group. Since both SLLN and CLT hold, provided a sufficiently large sample, X has an approximately normal distribution[21]. A permutation test can be performed to compare the mean value of each homology group. In the data of Figure 5.9 a permutation t-test (p-value 0.0028) differentiates the disk from the annulus in terms of the one-dimensional cycle.

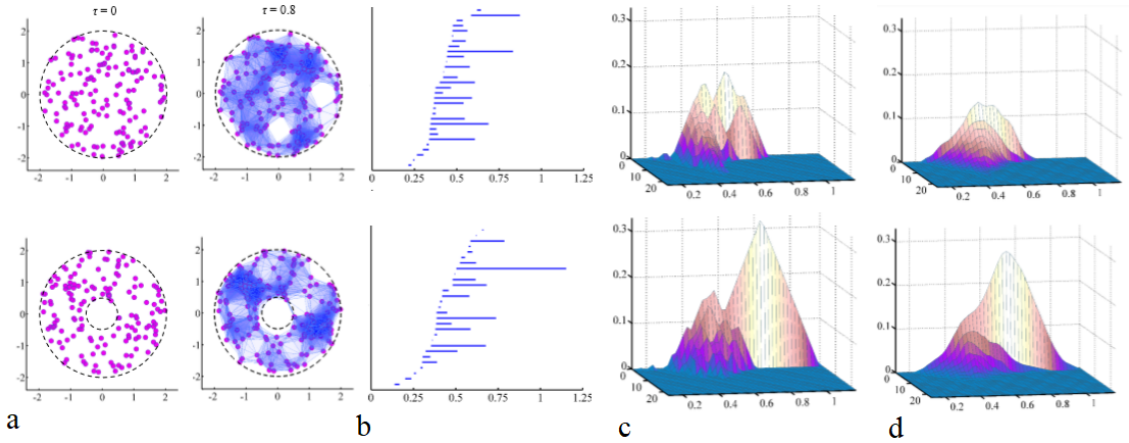


Figure 5.9: TDA on sets of points sampled for a disk and an annulus. (a) Some complexes of the two VR filtrations. (b) Their barcodes for the first homology group. (c) The PLs corresponding to each barcode. (d) The mean PLs. Image from [21].

We'll now derive the confidence sets for PLs using the CLT and the bootstrap method.

5.4.1 Central Limit Theorem

First we apply a functional to the persistence landscapes to obtain a real-valued random variable that satisfies the usual CLT.

Corollary 5.47. Assume $p \geq 2, E\|\Lambda\| < \infty$ and $E\left(\|\Lambda\|^2\right) < \infty$. For any $f \in L^q(\mathscr{S})$ with $\frac{1}{p} + \frac{1}{q} = 1$, let $Y = \int_{\mathscr{S}} f\Lambda = \|f\Lambda\|_1$. Then $\sqrt{n}\left[\overline{Y}_n - E(Y)\right] \xrightarrow{d} N(0, \mathrm{Var}(Y))$ where d denotes convergence in distribution and $N(\mu, \sigma^2)$ is the normal distribution with mean μ and variance σ^2.

Theorem 5.48 (Slutsky's Theorem)**.** *Let X_n, Y_n be sequences of random elements. If X_n converges in distribution to a random element X and Y_n converges in probability to a constant c, then*

- $X_n + Y_n \xrightarrow{d} X + c$;
- $X_n Y_n \xrightarrow{d} cX$;
- $X_n / Y_n \xrightarrow{d} X/c$.

where $\xrightarrow{d}$ denotes convergence in distribution.

Assume that $\lambda(X)$ satisfies the conditions of Corollary 5.47 and that Y is a corresponding real random variable. By Corollary 5.47 and Slutsky's Theorem we may use the normal distribution to obtain the approximate $(1-\alpha)$ confidence interval for $E(Y)$

$$\overline{Y}_n \pm z^* \frac{S_n}{\sqrt{n}}, \text{ where } S_n^2 = \frac{1}{n-1} \sum_{i=1}^{n} \left(Y_i - \overline{Y}_n\right)^2$$

and $z*$ is the upper $\dfrac{\alpha}{2}$ critical value for the normal distribution.

5.4.2 Bootstrap Method

Let the diagrams $\mathscr{P}_1, \ldots, \mathscr{P}_n$ be a sample from the distribution P over the space of persistence diagrams $\mathscr{D}_T$. Let $\mathscr{L}_1, \ldots, \mathscr{L}_n$ be the landscape functions corresponding to $\mathscr{P}_1, \ldots, \mathscr{P}_n$. In [12], the process $\sqrt{n}\left(\overline{\mathscr{L}}_n(t) - \mu(t)\right)$ was proved to converge to a Gaussian process, so the bootstrap empirical process can be used.

Let P_n be the empirical measure that the corresponding landscapes $\mathscr{L}_1^*, \ldots, \mathscr{L}_n^*$. Let $\overline{\mathscr{L}}_n^*$ be the empirical mean and $\hat{\theta}^* = \sup_{t \in \mathbb{R}} \left| \sqrt{n}\left(\overline{\mathscr{L}}_n^*(t) - \overline{\mathscr{L}}_n(t)\right)\right|$. Repeating this B times, we obtain $\hat{\theta}_1^*, \ldots \hat{\theta}_B^*$, and we compute the quantile q_α.

Theorem 5.49. *The interval $C_n(t)$ indexed by $t \in \mathbb{R}$, defined by*

$$C_n(t) = \left[\overline{\mathscr{L}}_n(t) - \frac{q_\alpha}{\sqrt{n}}, \overline{\mathscr{L}}_n(t) + \frac{q_\alpha}{\sqrt{n}}\right]$$

is a confidence band for $\mu(t)$

$$\lim_{n\to\infty} \mathbb{P}\left(\mu(t) \in C_n(t) \text{ for all } t\right) \geq 1 - \alpha$$

Example 5.50. Given the nine circles of radii 0.4 and 0.3 we obtain a sample $X_1, ..., X_{100}$ as follows: first, choose a circle C_i uniformly at random, then sample a point i.i.d. from it. Let $\mathscr{D}$ be the β_1 persistence diagram corresponding to the VR filtration for the sample, and $\mathscr{L}$ be the landscape corresponding to $\mathscr{D}$. We repeat this 50 times to obtain diagrams $\mathscr{D}_1, ..., \mathscr{D}_{50}$ and landscapes $\mathscr{L}_1, ..., \mathscr{L}_{50}$. Then, we use the bootstrap procedure to obtain the quantile $q_\alpha = 0.234$. Together with $\mathscr{L}_{50}$, this gives us an approximated 95% confidence band for $\mu(t) = E_P(\mathscr{L}_i(t))$. See Figure 5.10.

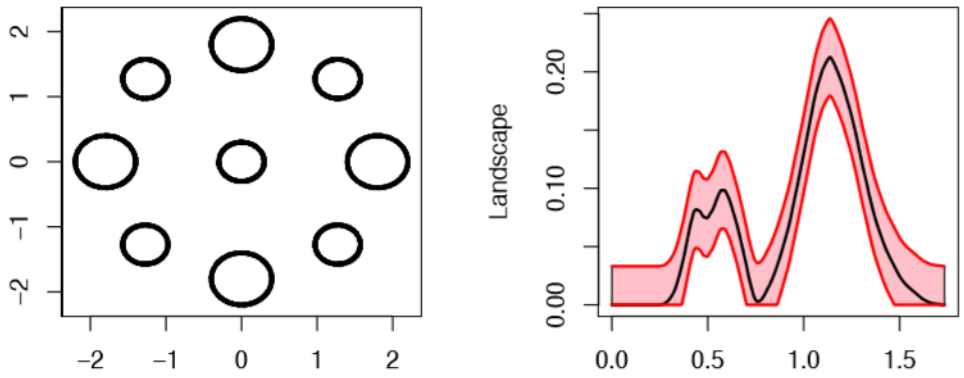


Figure 5.10: Left: The set of circles from which samples are taken. Right: The confidence band for the persistence landscape corresponding to the distance to the point set. Image from [22].

6. Implementation

In [74], different libraries such as javaPlex, Perseus, Dionysus, DIPHA, GUDHI, and PHAT were tested and compared. GUDHI[75], which is available for C++ and Python, has been highlight as one of the best available open-source libraries.

GUDHI proposed an efficient tree representation for simplicial complexes, the *simplex tree*, see Figure 6.1. The nodes of the tree are in bijection with the simplices of the complex.

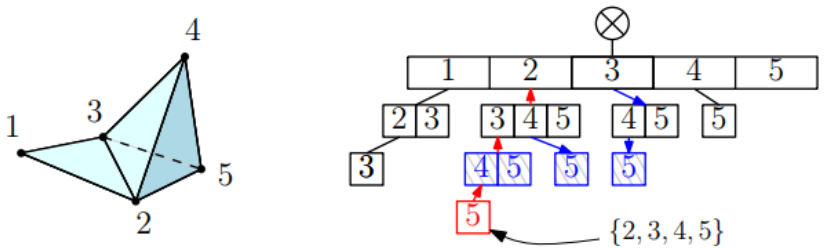


Figure 6.1: A simplicial complex and its representation as simplex tree. With focus on the simplex $\{2,3,4,5\}$. *Image from [16].*

The tree structure enables to store the informations of the complex and implement basic operations on it efficiently. Lots of interesting calculation on complexes can be performed, for example an elementary collapse of the free pair (τ,σ) consists in the removal of the two-nodes subtree containing the nodes representing τ and σ[16].

We'll now provide the implementation insights of some of the TDA steps previously mentioned.

6.1 Application

To analyse the topological information of different datasets a console application was implemented. In particular, the application:

- computes the VR filtrations of different numerical datasets (GUDHI - Python),
- extracts the persistences of the filtrations (GUDHI - Python),
- computes the confidence bands using the bootstrap method (TDA - Python),
- computes the persistence barcodes, the persistence diagrams and the Betti curves in a dynamic BI application that allows to easily access the insights provided by TDA (QlikView).

See Figure 6.2 for an overview of the upload process and Figure 6.3 for the TDA calculations steps.

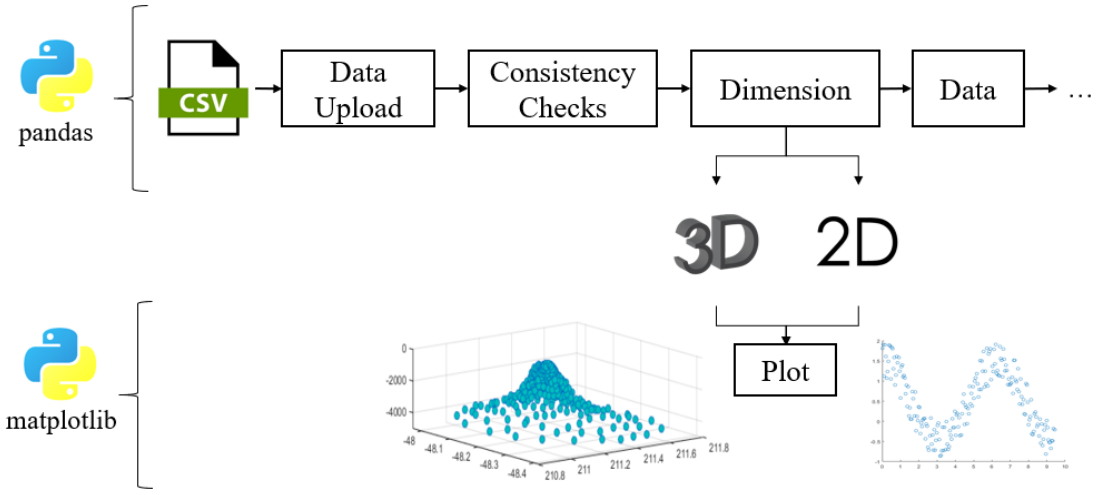


Figure 6.2: The data upload process: Pandas module was used to handle and check the input information and the Matplolib to achieve the 3d and 2d plots.

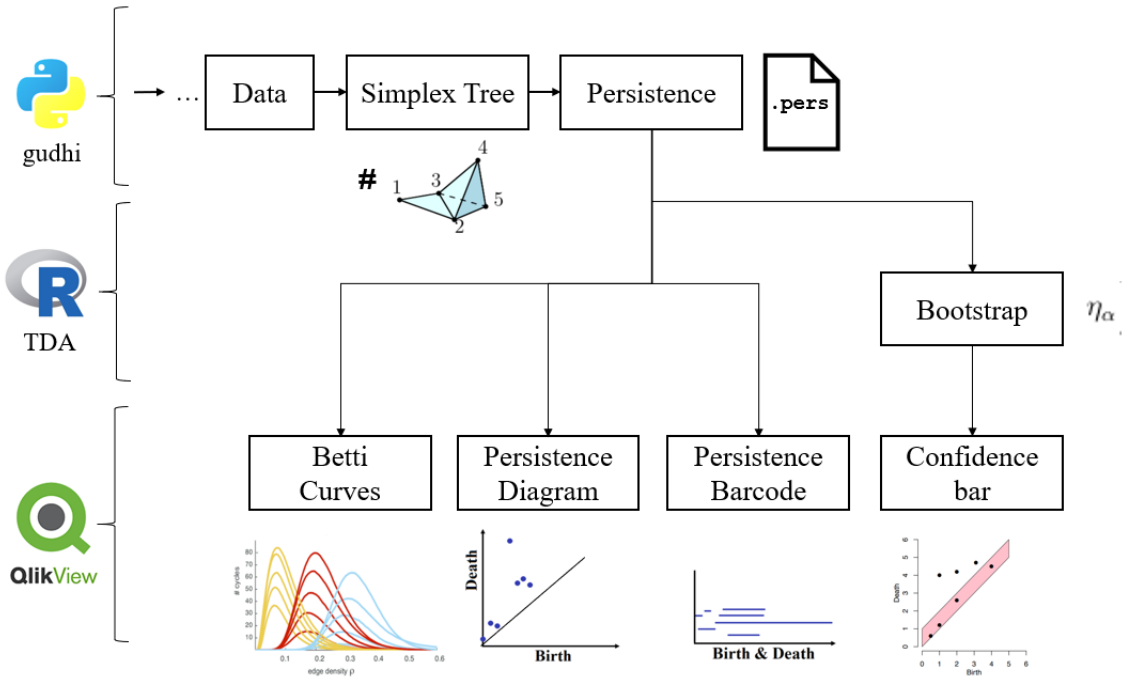


Figure 6.3: Persistence computation process.

The code of the console application is now provided.

6.1.1 Python

```
import os
import gudhi
from pathlib import Path
import subprocess
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
from pylab import *
import importlib
importlib.import_module('mpl_toolkits').__path__
from mpl_toolkits.mplot3d import Axes3D

def RaiseException(error):
        print(error)
        raise Exception(error)
```

```
def maxminDistance(points, dim):
        max_= 0
        min_= sys.maxsize
        avg_= 0
        N=0
        if dim==3:
                for point in points:
                        if str(point[0])!='nan' and str(point[1])!='nan' and str(
                            point[2])!='nan':
                                for point2 in points:
                                        if str(point2[0])!='nan' and str(point2[1])!='
                                            nan' and str(point2[2])!='nan' and point2!=
                                            point:
                                                a=math.pow(point[0]-point2[0],2)
                                                b=math.pow(point[1]-point2[1],2)
                                                c=math.pow(point[2]-point2[2],2)
                                                max_ = max(max_, sqrt(a+b+c))
                                                min_ = min(min_, sqrt(a+b+c))
                                                avg_=avg_+sqrt(a+b+c)
                                                N=N+1
        else:
                for point in points:
                        if str(point[0])!='nan' and str(point[1])!='nan' :
                                for point2 in points:
                                        if str(point2[0])!='nan' and str(point2[1])!='
                                            nan' and point2!=point:
                                                a=math.pow(point[0]-point2[0],2)
                                                b=math.pow(point[1]-point2[1],2)
                                                max_ = max(max_, sqrt(a+b))
                                                min_ = min(min_, sqrt(a+b))
                                                avg_=avg_+sqrt(a+b)
                                                N=N+1
        return max_,min_, avg_/N

def num_after_point(x):
    s = str(x)
    if not '.' in s:
```

```
        return 0
    return len(s) - s.index('.') - 1

def GetAxes(Name, dict):
        X=input('What column should be used as '+Name+ ' axes?\n')
        if X == '':
                RaiseException( Name+ ' axes must be valorized')
        try:
                X=int(X)
        except:
                RaiseException('This choice is not possible')
        if (X <= i and X > 0 and isinstance(X, int))==False:
                RaiseException('This choice is not possible')
        X = dict[str(X)]
        if np.issubdtype(df[X].dtype, np.number)==False:
                RaiseException(X +' is not a numerical column')
        return X

def GetDatasets(X, dict):
        if X == '':
                RaiseException( 'X axes must be valorized')
        try:
                X=int(X)
        except:
                RaiseException('This choice is not possible')
        if (X <= i and X > 0)==False:
                RaiseException('This choice is not possible')
        X = dict[str(X)]
        return X

def GetOptionalAxes(X, dict):
        try:
                X=int(X)
        except:
                RaiseException('This choice is not possible')
        if (X <= i and X > 0)==False:
                RaiseException('This choice is not possible')
```

```
        X = dict[str(X)]
        if np.issubdtype(df[X].dtype, np.number)==False:
                RaiseException(X +' is not a numerical column')
        return X


def drawPlot(plot, X, Y, Z):
        if plot=='y':
                fig = plt.figure()
                xs=[]
                ys=[]
                zs=[]
                if Z == '':
                        fig = plt.figure()
                        ax = fig.add_subplot(111)
                        for index, row in df.iterrows():
                                xs.append(row[X])
                                ys.append(row[Y])
                        ax.scatter(xs, ys)
                        ax.set_xlabel(X)
                        ax.set_ylabel(Y)
                        plt.show()
                        return xs, ys, None
                else:
                        fig = plt.figure()
                        ax = fig.add_subplot(111, projection='3d')
                        for index, row in df.iterrows():
                                xs.append(row[X])
                                ys.append(row[Y])
                                zs.append(row[Z])
                        ax.scatter(xs, ys, zs)
                        ax.set_xlabel(X)
                        ax.set_ylabel(Y)
                        ax.set_zlabel(Z)
                        plt.show()
                        return xs, ys, zs
        elif plot=='n':
                        xs=[]
```

```
                    ys=[]
                    zs=[]
                    if Z == '':
                            for index, row in df.iterrows():
                                    xs.append(row[X])
                                    ys.append(row[Y])
                            return xs, ys, None
                    else:
                            for index, row in df.iterrows():
                                    xs.append(row[X])
                                    ys.append(row[Y])
                                    zs.append(row[Z])
                            return xs, ys, zs
        else:
                RaiseException('Not valid option')

def AvgEuclideanDist(points):
        points_array=np.asarray(points)
        tot = 0

        for i in range(len(points_array)-1):
                tot += ((((points_array[i+1:]-points_array[i])**2).sum(1))**.5).
                    sum()

        avgEuclDist = tot/((points_array.shape[0]-1)*(points_array.shape[0])/2.)
        return avgEuclDist

if __name__ == '__main__':
        print('TDA summary application started')
        intervalPath = os.path.join(os.path.dirname(__file__), 'interval.txt')
        persistencePath = os.path.join(os.path.dirname(__file__), 'data.pers')
        bootstrapPath = os.path.join(os.path.dirname(__file__), 'bootstrap.csv')
        bootstrapCodePath = os.path.join(os.path.dirname(__file__), 'Boot.r')
        datasetPath= os.path.dirname(__file__)+'\\datasets'
        Restart= 'y'
        Norestart ='n'
        while Restart=='y':
```

```
Norestart='n'
while Norestart =='n':
        # Data upload
        print('Found datasets:')
        i=0
        dict={}
        for file in os.listdir(datasetPath):
                i=i+1
                dict[str(i)] = file;
                print(str(i)+') '+file)

        dataset = input("Which dataset should I use?\n")
        dataset= os.path.join(datasetPath, GetDatasets(dataset, dict
            ))

        # Conversion from CSV to Pandas dataframe
        df = pd.read_csv(dataset, error_bad_lines=False, sep=';')
        print('The attributes are: ')
        i=0
        dict={}
        for column in list(df.columns.values):
                i=i+1
                dict[str(i)] = column;
                print(str(i)+') '+column)

        if (df.columns.values==[]):
                RaiseException('The file has not the sufficient
                    number of attributes')
        dim=3

        # Attribute choice
        X = GetAxes('X', dict)
        Y = GetAxes('Y', dict)
        Z=input('What column should be used as Z axes? [facoltative
            ]\n')
        if Z == '':
                dim=2
```

```
        else:
                Z = GetOptionalAxes(Z, dict)

        # Plot
        plot = input("Do you want to plot the dataset? (y/n)\n")
        xs, ys, zs = drawPlot(plot, X, Y, Z)
        points=[]
        for elem in range(len(xs)):
                if dim ==3:
                        points.append([xs[elem], ys[elem], zs[elem]])
                if dim ==2:
                        points.append([xs[elem], ys[elem]])

        #Restart
        Norestart = input("Do you want to analyse this dataset? (y/n
            )\n")

#Average, Max and Min Euclidean distances
maxdist, mindist, avgdist= maxminDistance(points, dim)
#avgdist= AvgEuclideanDist(points)
print('The maximal Euclidean distance is '+str(maxdist))
print('The minimal Euclidean distance is '+str(mindist))
print('The average Euclidean distance is '+str(avgdist))

#Calculating VR Complex
maxdistc = input("Witch maximal proximity parameter use? \n")
print('Calculating VR complex using '+str(maxdistc)+ ' as proximity
     parameter')
rips_complex = gudhi.RipsComplex(points=points, max_edge_length=
    float(maxdistc))

#Building the Simplex Tree
print('Building the simplex tree')
simplex_tree = rips_complex.create_simplex_tree(max_dimension=int(
    dim))
print('Using this parameter VR complex has: \n' + repr(simplex_tree
    .num_vertices()) + ' vertices.\n'+ repr(simplex_tree.
```

```
    num_simplices()) + ' simplices.')

#Calculating Persistence
print('Calculating persistence')
diag = simplex_tree.persistence()
maxValue=0;
maxDecimal=0;
for interval in diag:
       if(interval[1][1]!= inf):
              maxValue=max(maxValue, interval[1][1])
              maxValue=max(maxValue, interval[1][0])
interval_ = np.linspace(0,maxValue,100)
with open(intervalPath, 'w') as f:
       for elem in interval_:
              f.write('%f\n' % elem)
print('Writing persistence')
simplex_tree.write_persistence_diagram(persistencePath)
print('Persistence written')

#Bootstrap
alpha = input("Which alpha use for bootstrapping?\n")
B = input("How many iterations for bootstrapping?\n")
print('Report info for Bootstrap')
with open(bootstrapPath, 'w') as f:
       f.write('Path;Dim;alpha;B;Columns;MaxPersistence'+'\n')
       f.write(str(dataset)+';'+str(dim)+';'+str(alpha)+';'+str(B)+
           ';'+str(X)+';'+str(maxdistc)+'\n')
       f.write(';;;;'+str(Y)+';\n')
       if (dim==3):
              f.write(';;;;'+str(Z)+';')
print('Compiling Bootstrap')
subprocess.call(["C:/Program Files/R/R-3.5.1/bin/R", '-f',
    bootstrapCodePath])
print('Program ended')
Restart = input("Do you want to analyze again? (y/n)\n")
```

Note that in Python, an R script is called to be processed. It computes the confidence band width and its code is provided below.

6.1.2 R

```
if (!require(package = "TDA")) {install.packages(pkgs = "TDA")}
library('TDA')
pathDataset<- read.csv("./bootstrap.csv", header = TRUE, sep=';',
    stringsAsFactors = FALSE)
B <- pathDataset[["B"]][1]
X <- pathDataset[["Columns"]][1]
Y <- pathDataset[["Columns"]][2]
S <- read.csv(pathDataset[["Path"]][1], header = TRUE, sep=';')
if(pathDataset[["Dim"]][1]==2) {
S <- as.matrix(S[,c(X, Y)])
XX <- as.matrix(S[,c(X)])
YY <- as.matrix(S[,c(Y)])
Xseq <- seq(XX[1], XX[2], by = ((XX[1]-XX[2])/(-10)))
Yseq <- seq(YY[1], YY[2], by = ((YY[1]-YY[2])/(-10)))
Grid <- expand.grid(Xseq, Yseq)
}
if(pathDataset[["Dim"]][1]==3) {
Z <- pathDataset[["Columns"]][3]
S <- as.matrix(S[,c(X, Y, Z)])
XX <- as.matrix(S[,c(X)])
YY <- as.matrix(S[,c(Y)])
ZZ <- as.matrix(S[,c(Z)])
Zseq <- seq(ZZ[1], ZZ[2], by = ((ZZ[1]-ZZ[2])/(-10)))
Xseq <- seq(XX[1], XX[2], by = ((XX[1]-XX[2])/(-10)))
Yseq <- seq(YY[1], YY[2], by = ((YY[1]-YY[2])/(-10)))
Grid <- expand.grid(Xseq, Yseq, Zseq)
}
h <- nrow(XX)^-.2
band <- bootstrapBand(X = S, FUN = kde, Grid = Grid, B = B, alpha =
    pathDataset[["alpha"]][1], h = B^-.2)
write.table(band[["width"]], "./confidence.txt" ,col.names = FALSE)
```

The visualizations achieved in QlikView using the proprietary language will be presented in the last chapter within the practical use of the program on some datasets.

7. Results

The program can be used on every numerical dataset having as separator a semicolon. We'll show the results found for different datasets:

- The Ecoli dataset (3D)
- Two datasets containing quite clear circular shapes (2D)
- Four datasets containing various elements. Two of them are corrupted by noise (2D and 3D)

Ecoli dataset

The *Ecoli Data Set*[55] containing protein localization sites information and having 336 istances has been analysed. Its attributes are:

1. Sequence Name: Accession number for the SWISS-PROT database (Categorical).
2. *mcg*: McGeoch's method for signal sequence recognition (Real).
3. *gvh*: von Heijne's method for signal sequence recognition (Real).
4. *lip*: von Heijne's Signal Peptidase II consensus sequence score (Binary).
5. *chg*: Presence of charge on N-terminus of predicted lipoproteins (Binary).
6. *aac*: score of discriminant analysis of the amino acid content of outer membrane and periplasmic proteins (Real).
7. *alm1*: score of the ALOM membrane spanning region prediction program (Real).
8. *alm2*: score of ALOM program after excluding putative cleavable signal regions from the sequence (Real).

Using the application we decided to consider the attributes *aac*, *mcg* and *gvh*. See Figure 7.1.

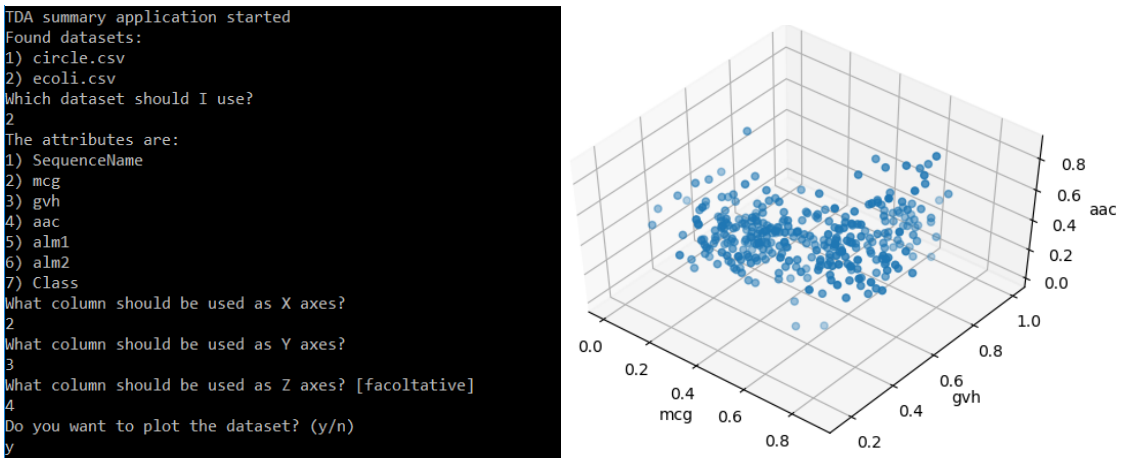


Figure 7.1: On the left, the first steps of the console applications used on the Ecoli dataset. On the right, the plot returned after the choise of plotting the dataset.

Between the points

- the average Euclidean distance is ~ 0.35.
- the maximal Euclidean distance is ~ 0.98.
- the minimal Euclidean distance is ~ 0.01.

We choose as maximal proximity parameter 0.3. If we generate a VR complex we get:

- 336 vertices
- $\sim$ 28 million simplices

The persistence computation, in the whole $[0, 0.3]$ interval of proximity parameters, revealed the presence of

- 336 components
- 135 loops
- 17 voids

In Figure 7.2 we can see the further steps of the application and the chosen parameters for the confidence band calculation: $\alpha = 0.05$ and 100 bootstrap iterations.

```
The maximal Euclidean distance is 0.9795407086997456
The minimal Euclidean distance is 0.009999999999999953
The average Euclidean distance is 0.3499458835863771
Witch maximal proximity parameter use?
0.3
Calculating VR complex using 0.3 as proximity parameter
Building the simplex tree
Using this parameter VR complex has:
336 vertices.
28116683 simplices.
Calculating persistence
Writing persistence
Persistence written
Which alpha use for bootstrapping?
0.05
How many iterations for bootstrapping?
100
```

Figure 7.2: Further steps of the console applications. The bootstrap parameters can be chosen.

All the information are stored in different files collected in QlikView.

```
SET DecimalSep='.';
SET MoneyThousandSep='.';

info:
LOAD Path, Dim as MaxDim, alpha, B, Columns, MaxPersistence
FROM [.\bootstrap.csv]
(txt, codepage is 1252, embedded labels, delimiter is ';', msq);

Qualify *;
persistenceUnique:
LOAD MaxPersistence resident info where MaxPersistence>0;
Unqualify *;
LET proximity = Peek('persistenceUnique.MaxPersistence');

interval:
LOAD @1 as proximity
FROM [.\interval.txt]
(txt, codepage is 1252, no labels, delimiter is '\t', msq);

confidence:
LOAD @1 as confidence, sqrt(2)*Num(@2) as band
FROM [.\confidence.txt]
(txt, codepage is 1252, no labels, delimiter is spaces, msq);

persistence:
LOAD
rowno() as uniqueIdentifier,
```

```
@1 as Dim,
@2 as start,
if(@3='inf', $(proximity), @3) as end
FROM [.\data.pers]
(txt, codepage is 1252, no labels, delimiter is '␣', msq);
```

The resulting application template is show in Figure 7.3. The persistence barcode, the persistence diagram and the Betti curves are provided.

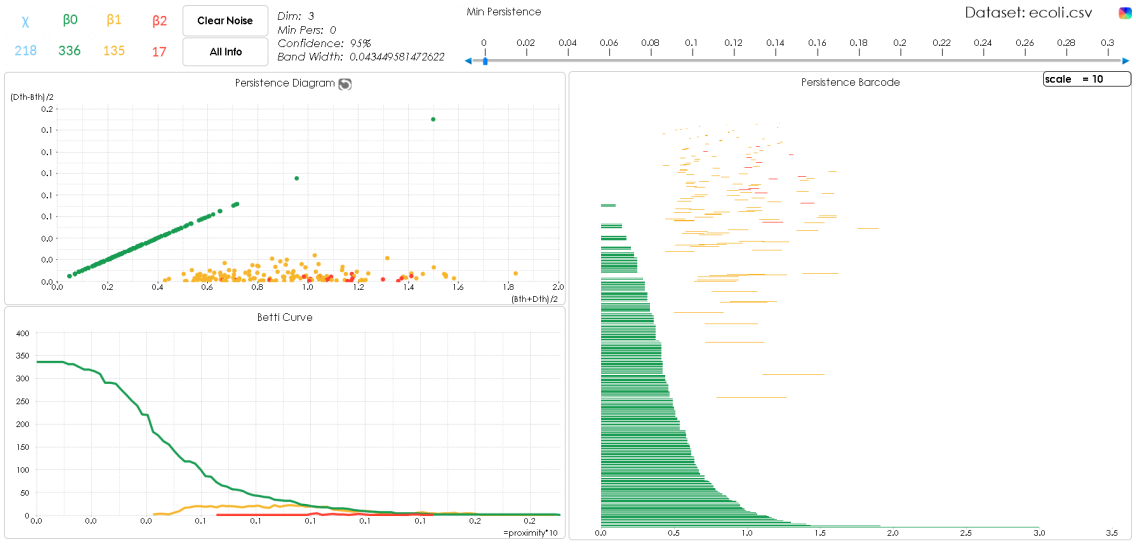


Figure 7.3: Application front-end template.

Some basic information are displayed, such as β_0, β_1, β_2 and the Euler characteristic. We can select a specific feature analysis by clicking on it. If "Clear Noise" is selected, bootstrapped noise is deleted, instead "Add All" sets the minimum considered persistence to 0. Using the scrollbar, the persistence threshold can be set to specific values. see Figure 7.4.

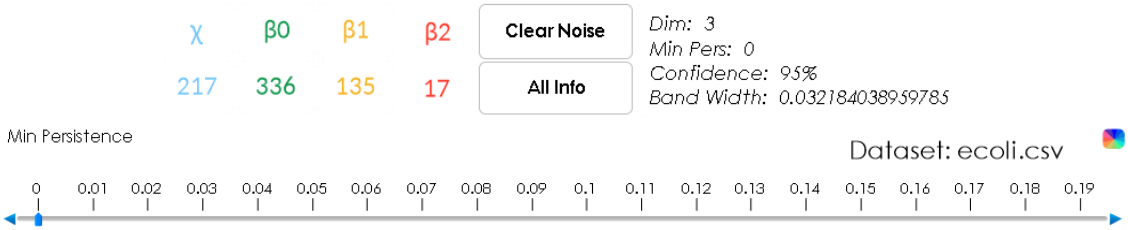


Figure 7.4: Buttons and KPIs of the TDA applications.

The specific dynamic selections are allowed, see Figure 7.5.

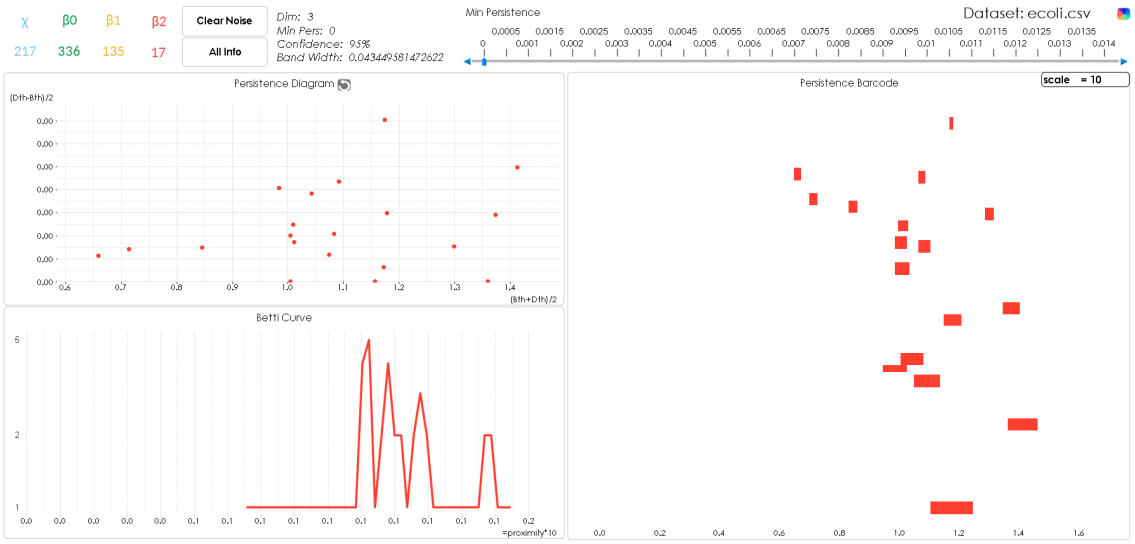


Figure 7.5: Application front-end template after pressing the red β_2 button.

Coming back to the analysis at hand, in Figure 7.7, 7.8 and 7.6 the topological summaries are presented. Clearing the persistence summaries from noise, we get 183 components and a loop. See Figure 7.9. The green elements refer to componets, the yellow ones to loops and the red ones to voids.

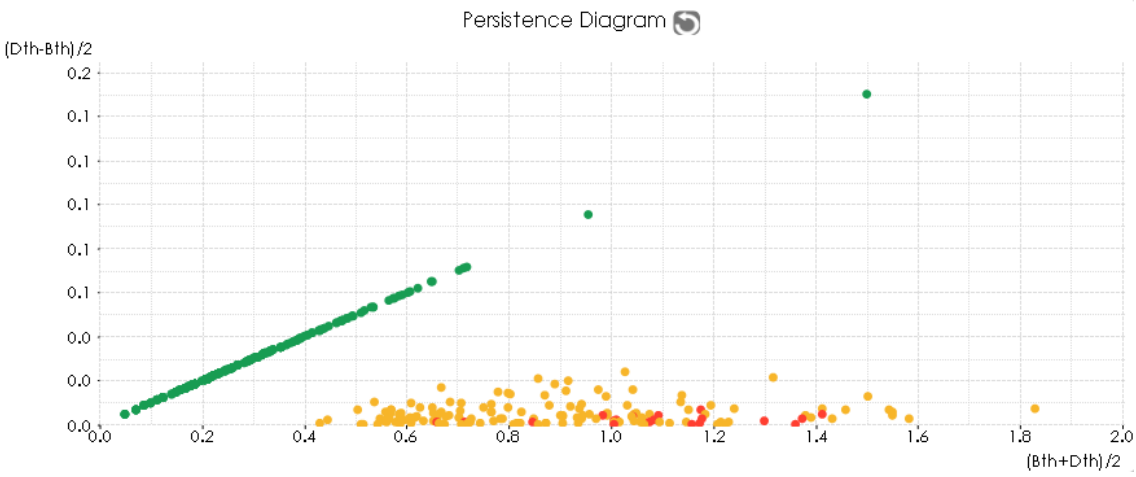


Figure 7.6: The rotated persistence diagram of Ecoli Data Set with the usual colors. The diagram can be de-rotated just by clicking the arrow image.

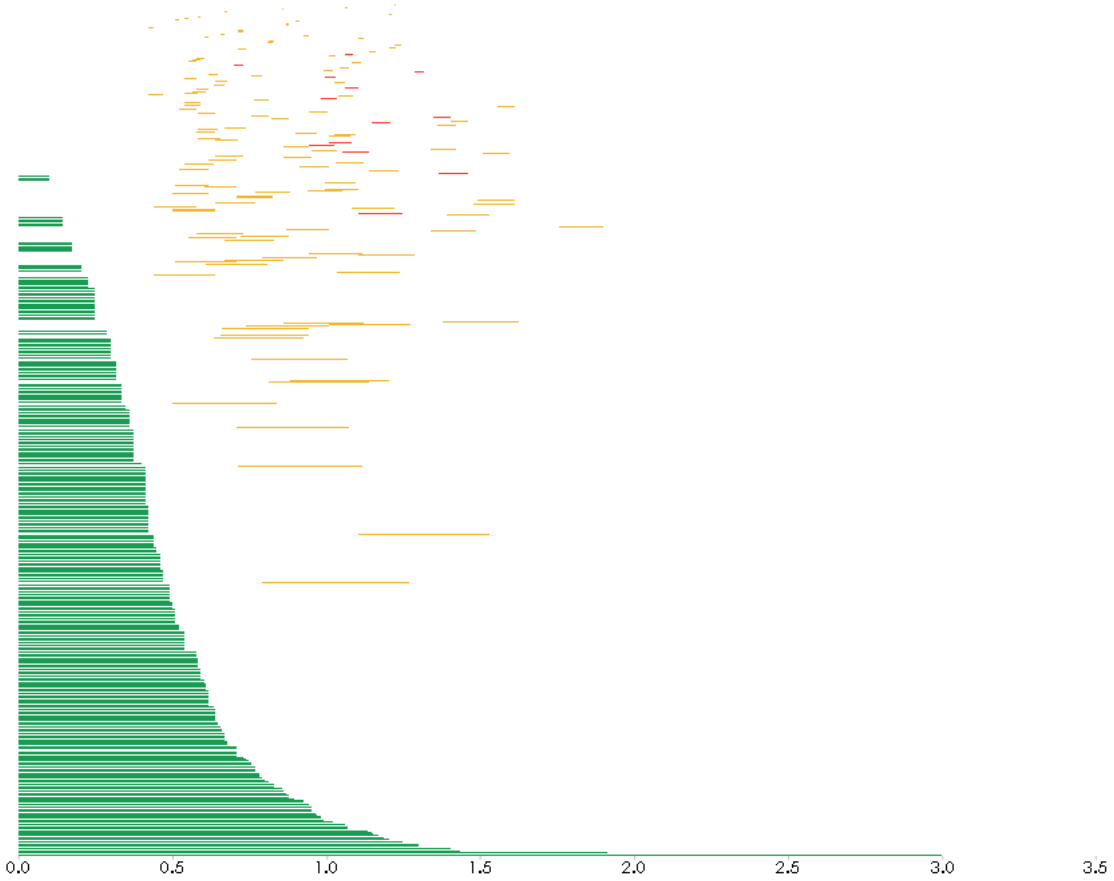


Figure 7.7: Persistence barcode of Ecoli Data Set ordered by persistence. The usual colors are used.

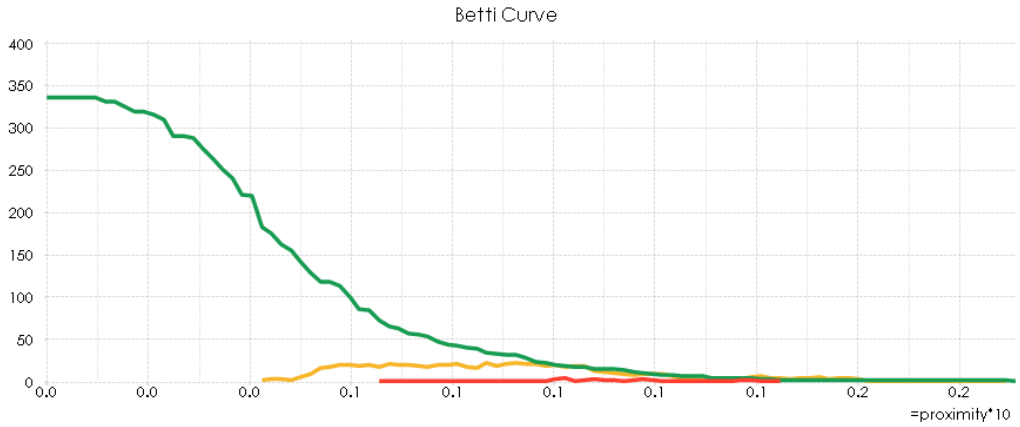


Figure 7.8: The Betti curves with the usual colors.

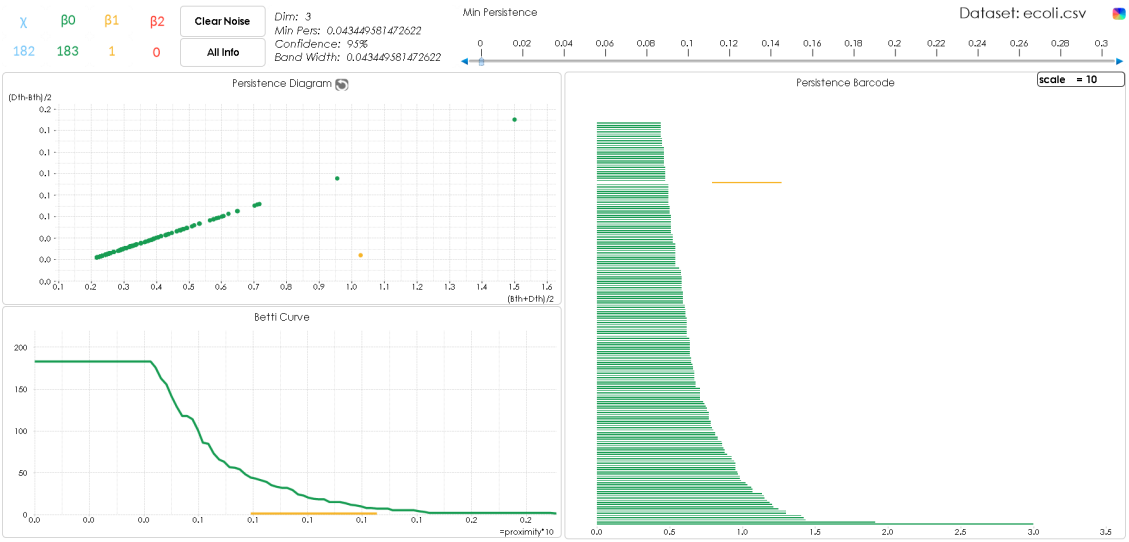


Figure 7.9: Persistence summaries of Ecoli Data Set after noise removal.

Circular Datasets

Consider the 2D-dataset represented in Figure 7.10.

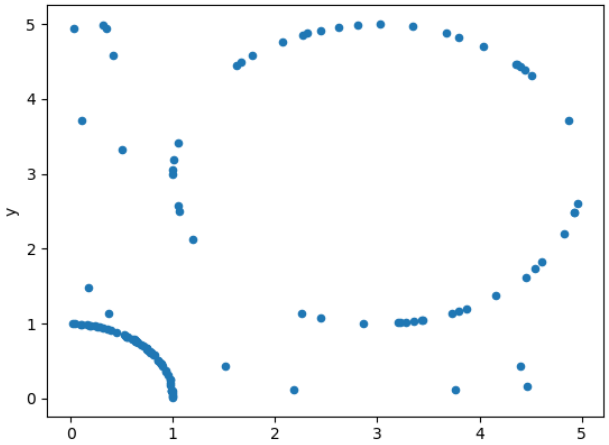


Figure 7.10: Noisy circle dataset scatterplot.

If we run our application setting $\alpha = 0.05$ and the number of bootstrap iterations to 100 we obtain the results in Figure 7.11.

It's clear that the circle is a relevant topological features despite the little noise. The Betti curves shows that the points collapse in a unique connected component with circular shape. If we analyse the dataset in Figure 7.12, we obtain the same result but from the charts in Figure 7.13 we can detect that first two persistent loops are found and they collapses in one that finally disappears.

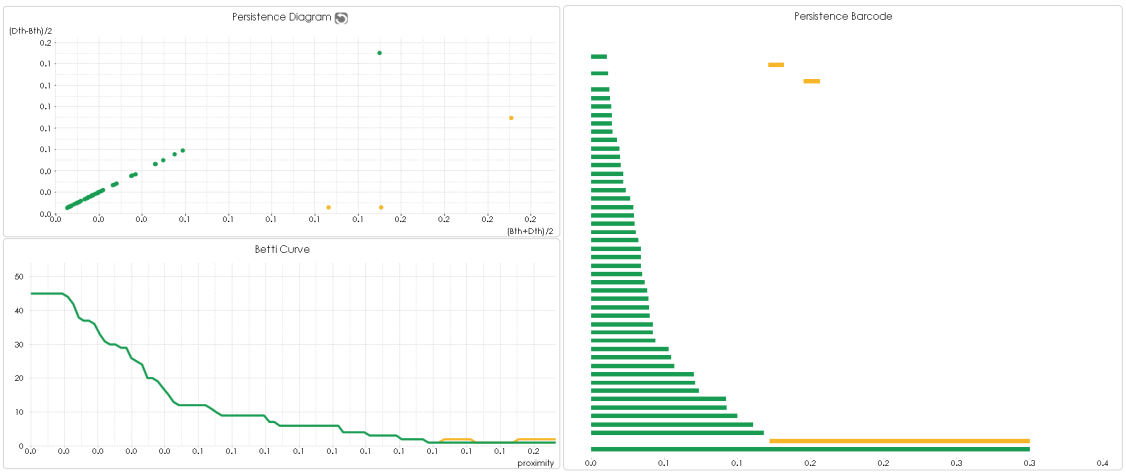


Figure 7.11: Topological summaries of the dataset of Figure 7.10.

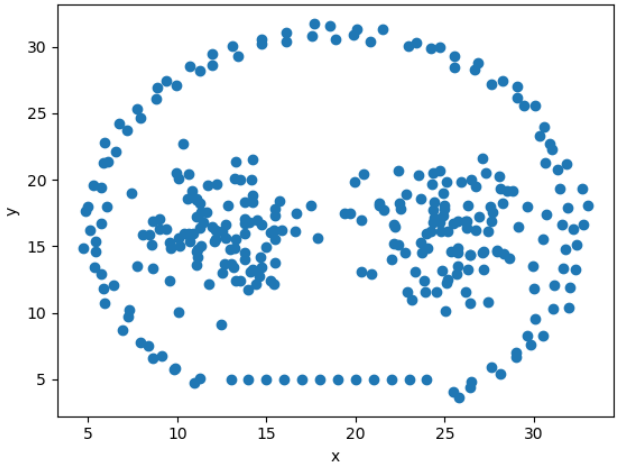


Figure 7.12: Circular dataset scatterplot.

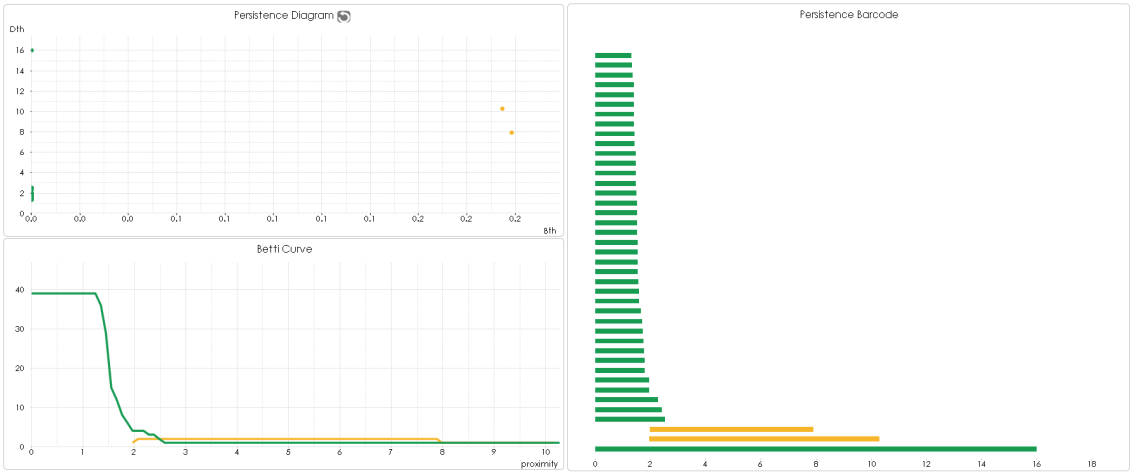


Figure 7.13: Topological summaries of the dataset of Figure 7.12.

Noisy Datasets

Finally, the datasets shown in Figure 7.14, 7.15, 7.16 and 7.17 were analysed and their topological summaries are shown on side. The more persistent loops and voids have been detected by the persistence barcode. The Python code to generate these datasets is provided.

```
#(...)
        with open(sphere, 'w') as f:
                f.write('x;y;z\n')
                for i in range(0,1000):
                        v = [0, 0, 0]
                        while np.linalg.norm(v) < .001:
                                x = np.random.randn()
                                y = np.random.randn()
                                z = np.random.randn()
                                v = [x, y, z]
                                v = v / np.linalg.norm(v)
                                f.write(str(round(v[0],2))+';'+str(round(v[1],2))+';'
                                    +str(round(v[2],2))+'\n')
```

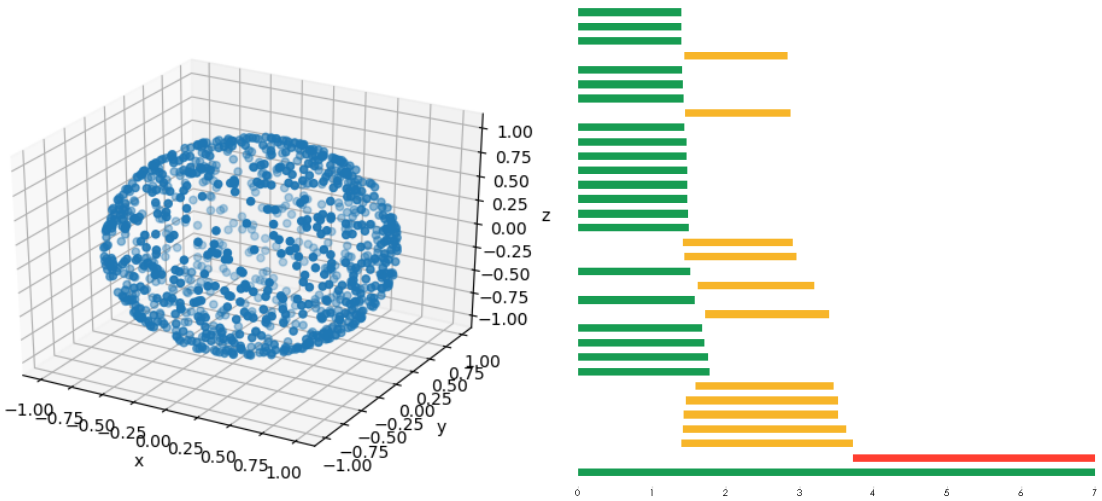


Figure 7.14: 3D sphere scatterplot on left and its barcode on right.

```
#(...)
        with open(sphere3, 'w') as f:
                f.write('x;y;z\n')
                for i in range(0,500):
                        v = [0, 0, 0]
```

```
            while np.linalg.norm(v) < .001:
                x = np.random.randn()
                y = np.random.randn()
                z = np.random.randn()
                v = [x, y, z]
                v = v / np.linalg.norm(v)
                f.write(str(round(v[0],2))+';'+str(round(v[1],2))+';'
                    +str(round(v[2],2))+'\n')
                f.write(str(round(v[0],2))+';'+str(round(v[1],2))+';'
                    +str(round(v[2],2)+0.7)+'\n')
                if (round(v[0],2)>0.6 or round(v[0],2)<-0.6) or (
                    round(v[1],2)>0.6 or round(v[1],2)<-0.6) :
                        f.write(str(round(v[0],2))+';'+str(round(v
                            [1],2))+';'+str(3)+'\n')
```

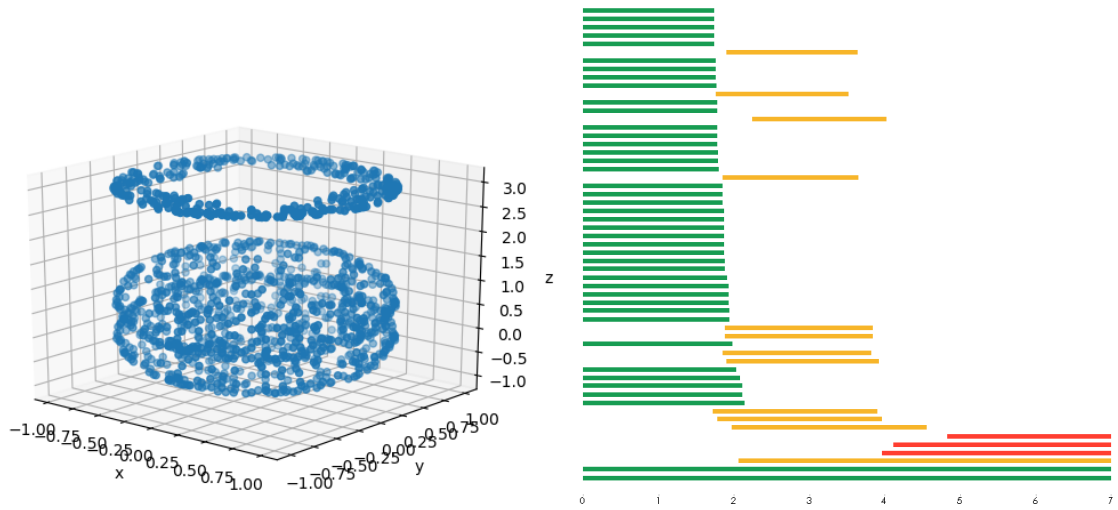


Figure 7.15: 3D dataset scatterplot on left and its barcode on right.

#(...)

```
        with open(spherenoise2d, 'w') as f:
            f.write('x;y;z\n')
            for i in range(0,200):
                v = [0, 0, 0]
                while np.linalg.norm(v) < .001:
                    x = np.random.randn()
                    y = np.random.randn()
                    z = np.random.randn()
                    v = [x, y, z]
```

```
                    v = v / np.linalg.norm(v)
                    f.write(str(round(x,2))+';'+str(round(y,2))+';'+str(
                        round(z,2))+'\n')
        for i in range(0,500):
            v = [0, 0, 0]
            while np.linalg.norm(v) < .001:
                x = np.random.randn()
                y = np.random.randn()
                z = np.random.randn()
                v = [x, y, z]
                v = v / np.linalg.norm(v)
                if (round(v[2],2)>0.6 or round(v[2],2)<-0.6) or (
                    round(v[1],2)>0.6 or round(v[1],2)<-0.6) :
                        f.write(str(0.3)+';'+str(round(v[1],2))+
                            ';'+str(round(v[2],2))+'\n')
                        f.write(str(0.3)+';'+str(round(v[1],2)
                            +0.5)+';'+str(round(v[2],2))+'\n')
                if (round(v[0],2)>0.6 or round(v[0],2)<-0.6) or (
                    round(v[1],2)>0.6 or round(v[1],2)<-0.6) :
                        f.write(str(round(v[0],2))+';'+str(round
                            (v[1],2))+';'+str(1.2)+'\n')
```

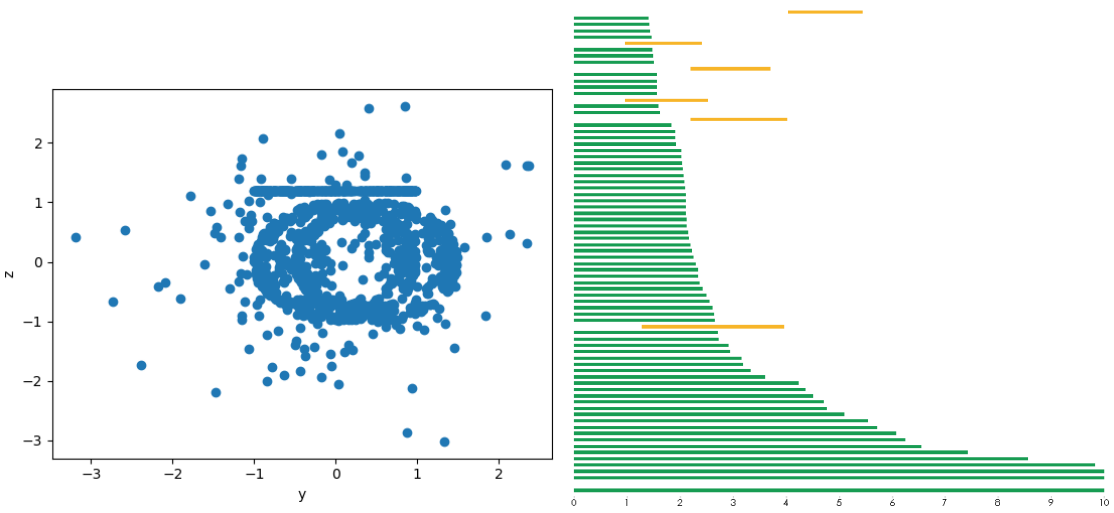


Figure 7.16: Noisy 2D dataset scatterplot on left and its barcode on right.

#(...)

```
    with open(spherenoise, 'w') as f:
        f.write('x;y;z\n')
```

```
for i in range(0,500):
        v = [0, 0, 0]
        while np.linalg.norm(v) < .001:
                x = np.random.randn()
                y = np.random.randn()
                z = np.random.randn()
                v = [x, y, z]
                v = v / np.linalg.norm(v)
                f.write(str(round(x,2))+';'+str(round(y,2))+';'+str(
                    round(z,2))+'\n')
                f.write(str(round(v[0],2))+';'+str(round(v[1],2))+';'
                    +str(round(v[2],2))+'\n')
                f.write(str(round(v[0],2))+';'+str(round(v[1],2))+';'
                    +str(round(v[2],2)+0.7)+'\n')
                if (round(v[0],2)>0.6 or round(v[0],2)<-0.6) or (
                    round(v[1],2)>0.6 or round(v[1],2)<-0.6) :
                        f.write(str(round(v[0],2))+';'+str(round(v
                            [1],2))+';'+str(3)+'\n')
```

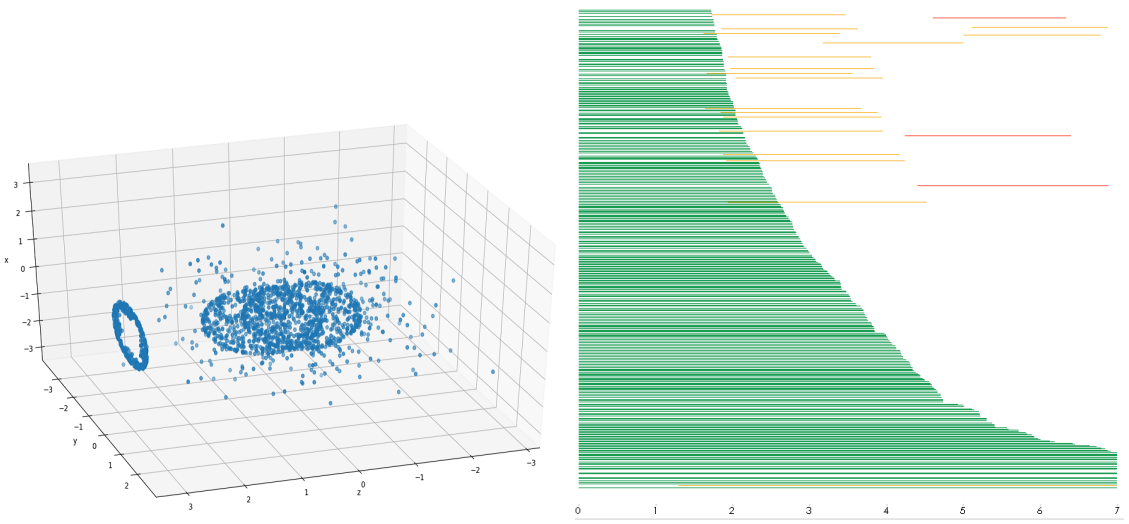


Figure 7.17: Noisy 3D dataset scatterplot on top and its barcode below.

List of Figures

List of Tables

Bibliography

[1] T. Abel, M. Alpaslan, M. A. Aragoon-Calvo, M. Cautun, R. van de Weygaert, B. Falck, J. E. Forero-Romero, R. Gonzalez, S. Gottloober, O. Hahn, W. A. Hellwing, Y. Hoffman, B. J. T. Jones, F. Kitaura, A. Knebe, N. I. Libeskind, S. Manti, M. Neyrinck, S. E. Nuza, N. Padilla, E. Platen, N. Ramachandra, A. Robotham, E. Saar, S. Shandarin, M. Steinmetz, R. S. Stoica, T. Sousbie, E. Tempel and G. Yepes, *Tracing the cosmic web*, Mon. Not. R. Astron. Soc. 473, 1195–1217, 2018.

[2] S. Ai, A. M. M Fadlallah, R. Kawai, I. Knowles, M. Nkashama and J. Wang, *Elliptic Equations And Systems With Nonlinear Boundary Conditions*, Disseration, University of Alabama at Birmingham, 2015.

[3] M. Alagappan, J. Carlsson, G. Carlsson, T. Ishkanov, A. Lehman, P. Y. Lum, G. Singh, and M. Vejdemo-Johansson, *Extracting insights from the shape of complex data using topology*, Scientific Reports volume 3, Article number: 1236, 2013.

[4] O. Alexandrov, *Illustration of connected sum*.

[5] D. Atsma, H. Bartelink, R. Bernards, H. Dai, L. Delahaye, M. J. van de Vijver, T. van der Velde, S. H. Friend, A. Glas, A. A. M. Hart, M. J. Marton, M. Parrish, J. L. Peterse, C. Roberts, S. Rodenhuis, E. T. Rutgers, G. J. Schreiber, L. J. van't Veer, D. W. Voskuil, A. Witteveen and D. H. Yudong, *A Gene-Expression Signature as a Predictor of Survival in Breast Cancer*, N Engl J Med 2002; 347:1999-2009, DOI: 10.1056/NEJMoa021967, 2002.

[6] Ayasdi, *Flagler Hospital*, on Ayasdi official site at link $https://s3.amazonaws.com/cdn.ayasdi.com/wp-content/uploads/2018/07/25070432/CS-Flagler-07.24.18.pdf$, 2018.

[7] Ayasdi, official site on $https://www.ayasdi.com/$ (01/09/2018).

[8] Ayasdi, *Anti-money Laundering* application example preview available on $https://www.ayasdi.com/applications/anti-money-laundering/$ (01/09/2018).

[9] Ayasdi, *Advanced Analytics in the Public Sector*, consulted on 10/10/2018 on $https://s3.amazonaws.com/cdn.ayasdi.com/wp-content/uploads/2015/02/13112032/wp-ayasdi-in-the-public-sector.pdf$, 2014.

[10] Ayasdi, president G. Carlsson, *TDA and Machine Learning: Better Together*, white paper, consulted on 01/10/2018.

[11] Ayasdi Core^{TM}, *Accelerating the Discovery of Powerful Insights from Complex Data*, consulted on $http://www.ayasdi.com/wp-content/uploads/2015/01/ayasdi-core.pdf$ on 01/10/2018.

[12] S. Balakrishnan, B. T. Fasy, F. Lecci, A. Rinaldo, A. Singh and L. Wasserman, *Confidence Sets For Persistence Diagrams* , DOI: 10.1214/14-AOS1252, 2014.

[13] K. E. Bassler, M. Boguna, G. Caldarelli, P. Colomer-de-Simo, M. M. Dankulov, A. Jamakovic, D. Krioukov P. Mahadevan, C. Orsini, Z. Toroczkai and A. Vahdat, *Quantifying randomness in real networks*, Nature Communications, 6:8627, DOI: 10.1038/ncomms9627, 2015.

[14] U. Bauer, *Algebraic perspectives of Persistence, The stability of persistence barcodes*, TUM, 2017.

[15] A. Benton, *A Brief Introduction to Computational Geometry*, University of Cambridge and Google UK Ltd.

[16] J. Boissonnat, C. Maria, *The Simplex Tree: An Efficient Data Structure for General Simplicial Complexes*, [Research Report] RR-7993, pp.20. <hal-00707901v1>, 2012.

[17] E.G.P. P. Bos, M. Caroli, R. van de Weygaert, H. Edelsbrunner, B. Eldering, M. van Engelen, J. Feldbrugge, E. ten Have, W. A. Hellwing, J. Hidding, B. J. T. Jones, N. Kruithof, C. Park, P. Pranav, M. Teillaud and G. Vegter, *Alpha, Betti and the Megaparsec Universe: on the Topology of the Cosmic Web*, arXiv:1306.3640v1 [astro-ph.CO], 2013.

[18] B. Brost, Supervisors: J. M. Møller and P. Winter, *Computing Persistent Homology via Discrete Morse Theory*, Thesis for the Master degree in Mathematics, Department of Mathematical Sciences, University of Copenhagen, 2013.

[19] A. Bronstein and M. Bronstein, *Numerical geometry of non-rigid objects, Metric model of shapes*, slides consulted on 05/05/2018 on $https://slideplayer.com/slide/4980986/$.

[20] A. Bronstein, M. Bronstein and R. Kimme, *Numerical geometry of non-rigid shapes* slides, SSVM, 2007.

[21] P. Bubenik, G. Heo, V. Kovacev-Nikolic, and D. Nikolic, *Using cycles in high dimensional data to analyze protein binding*, 2014.

[22] P. Bubenik, *Statistical Topological Data Analysis using Persistence Landscapes* , Journal of Machine Learning Research 16 (2015) 77-102, 2015.

[23] M. Candilera, Notes of *Omologia simpliciale e superficie reali*, Università degli studi di Padova, available on $http://www.math.unipd.it/\sim candiler/didafiles/pdf-files/homology.pdf$ and consulted on 01/04/2018.

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

[25] Z. Cang and G. Wei, *Topological fingerprints reveal protein-ligand binding mechanism*, arXiv:1703.10982v1 [q-bio.QM], 2017.

[26] G. Carlsson, *Why TDA and Clustering Are Not The Same Thing*, article on Ayasdi official site ($https://www.ayasdi.com/blog/machine-intelligence/why-tda-and-clustering-are-different/$), 2016.

[27] G. Carlsson, A.J. Levine, and M. Nicolau, *Topology based data analysis identifies a subgroup of breast cancers with a unique mutational profile and excellent survival*, Proc Natl Acad Sci U S A. 2011 Apr 26;108(17):7265-70. doi: 10.1073/pnas.110282610, 2011.

[28] G. Carlsson, *The Shape of Data* conference, Graduate School of Mathematical Sciences, University of Tokyo, 2015.

[29] G. Carlsson, *The Shape of Data* conference, Ayasdi Energy Summit, 2014.

[30] G. Carlsson, *Why Topological Data Analysis Works*, article on Ayasdi official site (*https* : *//www.ayasdi.com/blog/bigdata/why − topological − data − analysis − works/*), 2015.

[31] G. Carlsson, *Topology and data*, AMS Bulletin 46(2), 255–308, 2009.

[32] G. Carlsson, J. M. Chan and R. Rabadan, *Topology of viral evolution*, Proceedings of the National Academy of Sciences, 110 (46): 18566–18571, Bibcode:2013PNAS..11018566C. doi:10.1073/pnas.1313480110, ISSN 0027-8424, PMC 3831954. PMID 24170857, 2013.

[33] G. Carlsson, A. Collins,L. Guibas and A. Zomorodian, *Persistence Barcodes for Shapes*, Eurographics Symposium on Geometry Processing, Editors R. Scopigno and D. Zorin, 2004.

[34] G. Carlsson, T. Ishkhanov , D. L. Ringac, F. Memoli, G. Sapiro and G. Singh, *Topological analysis of population activity in visual cortex*, Journal of vision 8 8 (2008): 11.1-18.

[35] G. Carlsson, T. Ishkhanov, V. de Silva and A. Zomorodian, *On the Local Behavior of Spaces of Natural Images*, International Journal of Computer Vision, Volume 76, Issue 1, pp 1–12, 2008.

[36] G. Carlsson, T. Ishkhanov, F. Memoli, D. Ringach, G. Sapiro, *Topological analysis of the responses of neurons in V1*, preprint, 2007.

[37] G. Carlsson, A. J. Levine and M. Nicolau, *Topology-Based Data Analysis Identifies a Subgroup of Breast Cancers with a Unique Mutational Profile and Excellent Survival*, Proceedings of the National Academy of Sciences 108, no. 17: 7265–70, 2011.

[38] G. Carlsson and A. Zomorodian,*Computing persistent homology*, Discrete Comput.Geom., 33(2):249–274, 2005.

[39] N. J. Cavanna, M. Jahanseir and D. R. Sheehy, *A Geometric Perspective on Sparse Filtrations*, Computational Geometry, arXiv:1506.03797v1 [cs.CG], 2015.

[40] N. J. Cavanna, M. Jahanseir and D. R. Sheehy, *Visualizing sparse filtrations* video, 31st Symposium on Computational Geometry (Multimedia Session), 2015.

[41] P. Chardy, V. David and B. Sautour, *Fitting a predator–prey model to zooplankton time-series data in the Gironde estuary (France): Ecological significance of the parameters*, Estuarine, Coastal and Shelf Science, Volume 67, Issue 4, Pages 605-617, 2006.

[42] F. Chazal, *High-dimensional topological data analysis*, preliminary version (August 6, 2017), to appear in the Handbook of Discrete and Computational Geometry, J.E. Goodman, J. O'Rourke, and C. D. Tóth (editors), 3rd edition, CRC Press, Boca Raton, FL, 2017.

[43] F. Chazal, D. Cohen-Steiner, L. J. Guibas, F. Mémoli, S. Y. Oudot, *Gromov-Hausdorff Stable Signatures for Shapes using Persistence*, Computer Graphics Forum, 28 (5): 1393–1403, doi:10.1111/j.1467-8659.2009.01516.x, ISSN 1467-8659, 2009.

[44] F. Chazal, D. Cohen-Steiner and Q. Mérigot, *Boundary measures for geometric inference*, Found. Comp. Math., 10:221–240, 2010.

[45] F. Chazal, B. T. Fasy, F. Lecci, A. Rinaldo, and L. Wasserman, *Stochastic convergence of persistence landscapes and silhouettes*, Symposium on Computational Geometry (SoCG), 2014.

[46] F. Chazal, M. Glisse, C. Labruere and B. Michel, *Convergence rates for persistence diagram estimation in Topological Data Analysis*, Journal of Machine Learning Research, vol.16, pages 3603-3635, 2015.

[47] F. Chazal, M. Glisse, S. Oudot and V. de Silva, *The Structure and Stability of Persistence Modules*, SpringerBriefs in Mathematics, Springer International Publishing, 2016.

[48] F. Chazal and B. Michel, *An introduction to Topological Data Analysis: fundamental and practical aspects for data scientists*, arXiv:1710.04019v1 [math.ST], 2017.

[49] H. Chintakunta, H. Krim and A. C. Wilkerson, *Computing persistent features in big data: a distributed dimension reduction approach*, IEEE international conference on acoustics, speech and signal processing(ICASSP), pp11-15, 2014.

[50] M. K. Chung, H. Kang, B. Kim, H. Lee and D. S. Lee, *Persistent Brain Network Homology from the Perspective of Dendrogram*, DOI: 10.1109/TMI.2012.2219590, 2012.

[51] J. Cisewski-Kehea, S. B. Greenb, D. Nagai and X. Xu, *Finding cosmic voids and filament loops using topological data analysis*, arXiv:1811.08450v1 [astro-ph.CO], 2018.

[52] D. Cohen-Steiner, H. Edelsbrunner and J. Harer, *Stability of Persistence Diagrams*, in Discrete and Computational Geometry, vol. 37, p. 103-120, ftp://ftp-sop.inria.fr/prisme/dcohen/Papers/Stability.pdf, 2017.

[53] E. A. Coutsias, S. Martin, A. Thompson and J.P. Watson, *Topology of cyclo-octane energy landscape*, doi: 10.1063/1.3445267, 2010.

[54] B. Cottenceau, N. Delanoue, L. Jaulin, *Guaranteeing the homotopy type of a set defined bynon-linear inequalities*, Reliable Computing 13(5):381-398, DOI: 10.1007/s11155-007-9043-8, 2006.

[55] Dataset available on $https://archive.ics.uci.edu/ml/datasets/ecoli$.

[56] E. G. Escolar, Y. Hiraoka, A. Hirata, T. Nakamura, Y. Nishiura, *Persistent Homology and Many-Body Atomic Structure for Medium-Range Order in the Glass*, arXiv:1502.07445, 2015.

[57] H. Edelsbrunner, B. Eldering, W. A. Hellwing, M. L. Gavrilova, B. J. T. Jones, N. Kruithof, M. A. Mostafavi, C. Park, P. Pranav, G. Vegter, R. van de Weygaert and C. K. Tan, *Transactions on Computational Science XIV*, Berlin, Heidelberg: Springer-Verlag, pp. 60–101, ISBN 978-3-642-25248-8, 2011.

[58] J. Debosscher, T. V. Doorsselaere1 and H. Shariati, *Stellar Flares Observed in Long-cadence Data from the Kepler Mission*, The Astrophysical Journal Supplement Series, 232:26 (13pp), 2017.

[59] T. K. Dey, H. Edelsbrunner, S. Guha, and D. V. Nekhayev, *Topology preserving edge contraction*, Publications de l'Institut Mathematique, Beograd, 60:23–45, 1999.

[60] D. Du, *Contributions to Persistence Theory*, arXiv:1210.3092v4 [cs.CG], 2014.

[61] H. Edelsbrunner and J. Harer, *Computational Topology, An Introduction*, Departments of Computer Science and Mathematics, Duke University, DOI: 10.1007/978-3-540-33259-6 7 , 2010.

[62] H. Edelsbrunner, D. Letscher and A. Zomorodian, *Topological persistence and simplification*, Discrete Comput. Geom., 28:511–533, 2002.

[63] X. Feng, Y. Tong, G. W. Wei and K. Xia, *Persistent Homology for The Quantitative Prediction of Fullerene Stability*, arXiv:1412.2369v1 [q-bio.BM], 2014.

[64] X. Feng, Y. Tong, G. W. Wei and K. Xia, *Topological modeling of biomolecular data*, Nanyang Technological University.

[65] K. Fukumizu, Y. Hiraoka and G. Kusano, *Persistence weighted Gaussian kernel for topological data analysis*, 2016.

[66] P. A. Gagniuc, *Markov Chains: From Theory to Implementation and Experimentation*, USA, NJ: John Wiley and Sons, pp. 1-235, ISBN 978-1-119-38755-8, 2017.

[67] M. Gavrilova, J. T. Kenneth, M. A. Mostafavi, T*ransactions on Computational Science XIV. Special issue on Voronoi diagrams and Delaunay triangulation*, 10.1007/978-3-642-25249-5, 2011.

[68] R. Ghrist, *Three examples of applied and computational homology*, 2008.

[69] R. W. Ghrist, *Elementary applied topology*, 1st ed., United States, ISBN 9781502880857, OCLC 899283974, 2014.

[70] R. Ghrist, *Barcodes: The Persistent Topology Of Data*, Bull. Amer. Math. Soc. 45 (2008), 61-75 , Doi: https://doi.org/10.1090/S0273-0979-07-01191-3, 2007.

[71] R. Ghrist, *Homological Algebra and Data*, IAS/Park City Mathematics Series, S1079-5634(XX)0000-0, 2017.

[72] R. Ghrist and V. de Silva, *Coverage in sensor networks via persistent homology*, Algebraic and Geometric Topology, 7(1): 339-358, 2007.

[73] E. Gironi, advisor M. Ferri, co-advisor M. Barone, co-advisor I. Tomba, *Analisi prosodica della frase mediante omologia persistente*, Tesi di Laurea in Topologia Algebrica, Università di Bologna, 2017.

[74] P. Grindrod, H. A. Harrington, N. Otter, M. A. Porter and U. Tillmann, *A roadmap for the computation of persistent homology*, EPJ Data Science, 6:17 DOI10.1140/epjds/s13688-017-0109-5, 2017.

[75] GUDHI, *GUDHI Python module documentation* on $http://gudhi.gforge.inria.fr/python/latest/$, consulted on 01/10/2018.

[76] T. Halverson, C. M. Topaz and L. Ziegelmeier, *Topological Data Analysis of Biological Aggregation Models*, arXiv, DOI: 10.1371/journal.pone.0126383, 2014.

[77] L. Han, Y. Li, J. Liu, Y. Liu, Z. Liu, W. Nie, R. Wang and Z. Zhao, *PDB-wide collection of binding data: current status of the PDBbind database*, Bioinformatics, vol. 31, no. 3, pp. 405–412, 2015.

[78] Y. S. Han, J. C. Ye and J. Yoo, *Deep Residual Learning for Compressed Sensing CT Reconstruction via Persistent Homology Analysis*, Bio Imaging Signal Processing Lab, Korea Ad. Inst. of Science and Technology (KAIST), Korea, 2016.

[79] J. Harer and J. A. Perea, *Sliding Windows and Persistence: An Application of Topological Methods to Signal Analysis*, Foundations of Computational Mathematics, 15 (3): 799–838. doi:10.1007/s10208-014-9206-z, ISSN 1615-3375, 2014.

[80] H. A. Harrington, M. A. Porter and B. J. Stolz, *Persistent homology of time-dependent functional networks constructed from coupled time series*, DOI:10.1063/1.4978997, 2017.

[81] A. Hatcher, *Algebraic Topology*, Cambridge University Press, ISBN 0-521-79540-0, 2002.

[82] C. Hofer, .R. Kwitt, M. Niethammer, and A. Uhl, *Deep learning with topological signatures*, arXiv preprint arXiv:1707.04041, 2017.

[83] D. Horak, S. Maletić and M. Rajković, *Persistent homology of complex networks - IOPscience*, Journal of Statistical Mechanics: Theory and Experiment, 2009: P03034. arXiv:0811.2203, Bibcode:2009JSMTE..03..034H, doi:10.1088/1742-5468/2009/03/p03034, 2009.

[84] R. Ihaka and R. Gentleman, R Core Team, *R*, programming language and software environment available on $https://www.r-project.org/$ (01/09/2018).

[85] R. Jagadeesan and L. Sciarappa, Mentor A. Mathewo , *Simplicial Homology*, Fourth Annual MIT PRIMES Conference, 2014

[86] C. Keenan, *Discrete Differential Geometry*, *Reading 6 – Generalized Winding Numbers*, Carnegie Mellon University, CS 15-458/858B, 2016.

[87] H. E. Kim, *Evaluating Ayasdi's Topological Data Analysis For Big Data*, Master Thesis, Advisors S. Trahasch and R. V. Zicari, Offenburg University of Applied Sciences Goethe University Frankfurt, 2015.

[88] R. Kraft, *Illustrations of Data Analysis Using the Mapper Algorithm and Persistent Homology*, Degree Project In Applied And Computational Mathematics, Kth Royal Institute Of Tcehnology, Sweden, 2016.

[89] M. R. Kosorok, *Introduction to Empirical Processes and Semiparametric Inference*, ISBN 978-0-387-74978-5, 2008.

[90] R. Kraft, *Illustrations of Data Analysis Using the Mapper Algorithm and Persistent Homology*, 2016.

[91] V. Kurlin, *A one-dimensional Homologically Persistent Skeleton of an unstructured point cloud in any metric space*, Computer Graphics Forum (CGF), 34 (5): 253–262. doi:10.1111/cgf.12713, 2015.

[92] V. Kurlin, *A fast and robust algorithm to count topologically persistent holes in noisy clouds*, IEEE Conference on Computer Vision and Pattern Recognition (CVPR), doi:10.1109/CVPR.2014.189, 2014.

[93] V. Kurlin, *A Homologically Persistent Skeleton is a fast and robust descriptor of interest points in 2D images*, Lecture Notes in Computer Science (Proceedings of CAIP: Computer Analysis of Images and Patterns), 9256: 606–617, doi:10.1007/978-3-319-23192-1_51, 2015.

[94] P. Lambrechts, *The Poincaré conjecture and the shape of the universe* slides, Wellesley College, 2009.

[95] M. Lesnick, *Studying the Shape of Data Using Topology*, Institute for Advanced Study, School of Mathematics official site $https://www.ias.edu/ideas/2013/lesnick-topological-data-analysis$, 2013.

[96] M. Lotz, *Persistent Homology For Low-complexity Models*, Manchester Institute for Mathematical Sciences, The University of Manchester, ISSN 1749-9097, 2017.

[97] R. MacPherson and B. Schweinhart, *Measuring shape with topology*, Journal of Mathematical Physics, 53 (7): 073516. Bibcode:2012JMP....53g3516M, doi:10.1063/1.4737391, ISSN 0022-2488, 2012.

[98] S. Maletić, M. Rajković and Y. Zhao, *Persistent topological features of dynamical systems*, doi.org/10.1063/1.4949472, 2016.

[99] A. L. Mamuye, E. Merell, M. Piangerelli, M. Quadrini, M. Rucco, and L. Tesei, *Survey of TOPDRIM applications of Topological Data Analysis*, University of Camerino, Camerino, Italy, 2016.

[100] C. Maria, *Algorithms and data structures in computational topology*, Université Nice Sophia Antipolis, 2014.

[101] S. Martin and J.P. Watson, *Non-manifold surface reconstruction from high-dimensional point cloud data*, Computational Geometry, Volume 44, Issue 8, pp 427-441, 2011.

[102] K. Mischaikow, V. Nanda, *Morse theory for filtrations and efficient computation of persistent homology*, Discrete Comput Geom 50:330-353, 2013.

[103] V. Nanda and R. Sazdanovic, *Simplicial models and topological inference in biological systems*, In Discrete and Topological Models in Molecular Biology, pages 109–141. Springer, 2014.

[104] M. Offroy, *Topological data analysis: A promising big data exploration tool in biology, analytical chemistry and physical chemistry*, Analytica Chimica Acta, 910: 1–11. doi:10.1016/j.aca.2015.12.037, 2016.

[105] A. Patania, G. Petri and F. Vaccarino, *Topological analysis of data*, EPJ Data Sci. (2017) 6: 7, $https://doi.org/10.1140/epjds/s13688-017-0104-x$, 2017.

[106] A. Patania, G. Petri, F. Vaccarino and J.G. Young, *Construction of and efficient sampling from the simplicial configuration model*, arXiv:1705.10298v2 [physics.soc-ph], 2017.

[107] Qlik, *QlikView*, business intelligence software available on $https://www.qlik.com/us/products/qlikview$ (01/09/2018).

[108] J. Pellerin, *Accounting for the geometrical complexity of geological structural models in Voronoi-based meshing methods*, DOI: 10.13140/RG.2.1.2719.2169, 2014.

[109] C. C. Pinter, *A Book of Abstract Algebra*, Second Edition, Chap. 12, Dover Publications, Inc., Mineola, New York, 1982.

[110] G. van Rossum, Python Software Foundation, *Python*, interpreted high-level programming language available on $https://www.python.org/$ (01/09/2018).

[111] R. Roesler, *Non-Euclidean Geometry*, available on $http://www.compadre.org/profiles/postfiles/RoeslerNon-Euclidean.pdf$ (01/09/2018).

[112] K. Rykaczewski, K. Stencel and P. Wiśniewski, *An Algorithmic Way to Generate Simplexes for Topological Data Analysis*, Conference: 25th international Workshop on Concurrency, Specification and Programming, 2016.

[113] P. Rogers, *Big Data is a Three Part Harmony*, on Ayasdi official site at link $https://www.ayasdi.com/blog/bigdata/big-data-is-a-three-part-harmony/$, 2014.

[114] D. Salinas, *The Gudhi library: Simplification of Simplicial Complexes* slides, Gudhi workshop, 2014.

[115] Y. Seth, *Bootstrapping – A Powerful Resampling Method in Statistics*, on $https://yashuseth.blog/2017/12/02/bootstrapping-a-resampling-method-in-statistics/$, 2017.

[116] D. Sorrells, *Homology Groups And Persistence Homology* slides on $https://slideplayer.com/slide/4042965/$, 2015.

[117] D. Taylor, *Topological data analysis of contagion maps for examining spreading processes on networks*, Nature Communications, 6 (6): 7723, arXiv:1408.1168, Bibcode:2015NatCo...6E7723T, doi:10.1038/ncomms8723, ISSN 2041-1723, 2015.

[118] J. Tierny, *Introduction to Topological Data Analysis*, Sorbonne Universités, UPMC Univ Paris, Laboratoire d'Informatique de Paris, available on $https://www-pequan.lip6.fr/tierny/stuff/teaching/tierny_opologicalDataAnalysis.pdf$ (01/09/2018).

[119] Y. Umeda, *Time series classification via topological data analysis*, Transactions of the Japanese Society for Artificial Intelligence, 32(3):D-G72.1, 2017.

[120] Y. Umeda, *Time Series Classification via Topological Data Analysis*, Fujitzu Laboratories Ldt., 2016.

[121] K. G. Wang, *The Basic Theory of Presisten Homology*, 2012.

[122] B. Wang, G. Wei, *Objective-oriented Persistent Homology*, arXiv:1412.2368v1 [q-bio.BM], 2014.

[123] G. W. Wei and K. Xia, *Persistent homology analysis of protein structure, flexibility and folding*, arXiv:1412.2779v1 [q-bio.BM], 2014.

[124] G. H. Winslow, *Classification of Compact 2-manifolds*, Theses and Dissertations, Virginia Commonwealth University, 2016.

[125] M. Wright, *Introduction to Persistent Homology* video on M. Wright channel $https://www.youtube.com/watch?v=2PSqWBIrn90$ consulted on 10/10/2018, 2016.

[126] A. J. Zomorodian, *Topology for Computing*, Stanford University, Cambridge University Press, 2005.