UNIVERSITAT DE BARCELONA

Facultat de Matemàtiques
i Informàtica

GRAU DE MATEMÀTIQUES

Treball final de grau

An application of the Mapper algorithm to resynchronization in a model of non-ischaemic cardiomyopathy

Autor: Joan Guich Estevez

Directors: Carles Casacuberta Vergés
Aina Ferrà Marcús

Realitzat a: Departament de Matemàtiques i Informàtica

Barcelona, 13 de juny de 2022

Contents

Abstract

Mapper is one of the principal tools in topological data analysis (*TDA*) that enables studying topological features of high-dimensional datasets. Many studies from different fields, such as medicine and sports, have recently applied the Mapper algorithm to extract outstanding information from data.

In this work, our goal is to substantiate the conclusions from *Comparison between endocardial and epicardial cardiac resynchronization in an experimental model of non-ischaemic cardiomyopathy* study. In particular, we look for the optimal heart region where cardiac resynchronisation therapy offers a better result. Even though the core of the study is practical, we also profoundly study the theory behind the Mapper algorithm and the statistical methods we apply throughout the process.

Resum

Mapper és un dels mètodes principals dins de la branca de *topological data analysis* (*TDA*) que permet estudiar característiques topològiques sobre conjunts de dades de grans dimensions. Recentment, molts estudis de diferents àrees, com la medicina o els esports, han aplicat l'algorisme de Mapper per extreure informació rellevant de les dades tractades.

En aquest treball, el nostre objectiu és refermar les conclusions que es van obtenir a l'estudi *Comparison between endocardial and epicardial cardiac resynchronization in an experimental model of non-ischaemic cardiomyopathy*. En concret, volem aconseguir la posició del cor òptima on una teràpia de resincronització cardíaca tingui més eficiència. Malgrat que la part principal d'aquest estudi és pràctica, també profunditzem en la teoria que hi ha al darrere de l'algorisme de Mapper i els mètodes estadístics utilitzats durant el procés.

2020 Mathematics Subject Classification. 55N31, 55P10, 62P10

Agraïments

Abans d'entrar en matèria, m'agradaria dedicar un apartat d'aquest treball a donar agraïments a totes aquelles persones que m'han recolzat durant el transcurs d'aquests mesos.

En primer lloc, donar les gràcies al Dr. Carles Casacuberta i a l'Aina Ferrà per oferir-me l'oportunitat de formar part d'aquest projecte. A més a més, els vull agrair la seva predisposició a compartir el seu coneixement i guiar-me durant tot el procés. Ha estat un plaer poder treballar al vostre costat.

Així mateix, pel suport incondicional que sempre ofereixen i la seva paciència, dono les gràcies als meus pares, en Pau i a les meves dues àvies. Sempre us estaré agraït.

Finalment, vull agrair als els meus amics i amigues per recolzar-me i animar-me en els moments més difícils al llarg d'aquest semestre. En especial, Ivan, Nofre, Álvaro, Gemma, Ton, Bernat, Miquel i Oscar, sense vosaltres hauria sigut molt complicat continuar endavant.

Introduction

These notes were born from the idea to support the results obtained in the paper *Comparison between endocardial and epicardial cardiac resynchronization in an experimental model of non-ischaemic cardiomyopathy* carried out by a team from the Department of Cardiology in Hospital de la Santa Creu i Sant Pau. The study aimed to *"compare the acute response of biventricular pacing from the LV epicardium and endocardium in a swine non-ischaemic cardiomyopathy (NICM) model of dyssynchron"*. In other words, they wanted to find differences, or relevant information, by comparing the obtained heart testings values between a swine with or without a bipolar pacing electrode.

They tested this method on six different swine individuals. The values obtained for each swine were differentiated by the different heart regions and also by endocardial or epicardial pacing. There are three heart regions: basal, mid, and apical. Then, their results reflected that the pacing from basal regions, either from the epicardium or endocardium, produced better responses than mid or apical regions. On the other hand, they could not find any relevant information about the comparison between endocardial and epicardial pacing. After applying traditional statistical methods to their dataset, a whole set of conclusions was abstracted.

Afterwards, the Faculty of Mathematics and Computer Science of the Universitat de Barcelona joined the study in order to apply topological data analysis techniques to the same dataset. The principal aim of this collaboration was to reaffirm the results referred to the distinction between heart regions' responses. On the other hand, we were open to the possibility of finding a differentiation between the endocardium and epicardium.

Topological data analysis is a branch of applied mathematics that uses topological techniques and concepts to analyse data. Topological data analysis, commonly abbreviated by TDA, was born from the necessity to analyse high-dimensional data that traditional statistical methods could not manage.

During the last years, the collection of data in almost every area of our society has been growing exponentially. This vast amount of data plays a fundamental role

in our lives since we can abstract determinant and valuable information from it. However, there was a point in the past when we found ourselves in a situation where we had to deal with high-dimensional datasets that we could not analyse through the traditional methods we used to apply.

The principal aim of TDA is to find relevant information about the studied dataset through quantitative and qualitative topological features (e.g., clusters, branches, holes...). Intuitively, TDA tries to extract information from the shape of data.

This branch of applied mathematics is a continually growing area, and nowadays there is much investment in it. One principal reason for this significant investment is the relevant and valuable results from real-life studies. In particular, there are many examples applied to the medical field; an example could be the paper "*Identification of type 2 diabetes subgroups through topological analysis of patient similarity*" [1]. As the title can tell, they identified three distinct subgroups of Type 2 diabetes (T2D) from topology-based patient-patient networks.

Nowadays, the two principal methods used in TDA are the Mapper algorithm [Singh et al., 2007] and persistent homology.

In this paper, we apply the Mapper algorithm to study the topological features from the Hospital de Sant Pau dataset. We divide this paper into two sections. In the first one, we give a fully detailed explanation of the theoretical concepts behind Mapper and other tools we have used to analyse the data. On the other hand, the second part states the procedure and results from applying the Mapper algorithm to the given dataset.

The idea of the Mapper algorithm is, given a data set X and a well-chosen real-valued function $f : X \to \mathbb{R}^d$, to summarise X through the nerve of the refined pullback of a cover $\mathcal{U}$ of $f(X)$. For well-chosen covers $\mathcal{U}$, this nerve is a graph providing an easy and convenient way to visualise a summary of the data.

Hence, in the first section of these notes (the theoretical one), we state the necessary theory to understand the nerve concept and one of the most important theorems in TDA, the Nerve Theorem. This theorem states the relation between a nerve and the respective topological space through topological features. Furthermore, we also introduce the concepts for principal components analysis (PCA) and give a short comparison between PCA and singular value decomposition. Finally, at the end of this section, we briefly discuss the stability of 1-dimensional Mapper.

In the study of the data, firstly, we briefly introduce a complete description of the dataset we use to facilitate the reader's comprehension of everything implemented and abstracted from it. Then, we discuss the results obtained after submitting our

data to a filter function and PCA analysis and claim that the tools that we used were optimal for our study. After visualizing our point cloud, we apply the Mapper algorithm to it and present the outputs. To conclude the paper, we recapitulate, observe and examine all the results that have been obtained throughout the study, and take out some final conclusions.

Chapter 1

Nerve Theorem

In this section, we will study some theoretical concepts behind Topological Data Analysis, *TDA*, and the Mapper algorithm. The goal of *TDA* is to make meaningful signatures of the data through topological tools. Hence, these signatures lead to topological invariants, which enable a greater understanding of the relationships in —and transformations of— data. Moreover, these techniques allow us to obtain some information and conclusion from the data set that we could not acquire using traditional methods.

The Mapper algorithm uses the concept of nerve of a cover of a topological space X. Moreover, we also study the well-known Nerve Theorem. This theorem states the relation of topological features between a topological space X and the nerve of a suitable cover of X. This theorem has different versions, but we will only state the homotopical and homological versions. Since we will prove the result for the homotopical version, we have to study all the theoretical concepts to understand it. On the other hand, we just introduce the theorem itself and the respective proof for the homological result.

Even though the nerve of a given dataset is the principal concept of the Mapper algorithm, we will see that, in general, it does not satisfy the Nerve Theorem. However, at the end of this section we state some results related to the algorithm's stability (or instability). This stability is explained through the perturbation of the parameters that have to be selected by the user.

1.1 Homotopical Nerve Theorem

We begin by studying the homotopical version of the Nerve Theorem and the necessary theoretical background to understand and prove it.

Definition 1.1. Given a subset X of a topological space, we define an *open cover* as a collection of open subsets $\mathcal{U} = \{U_i\}_{i \in I}$ such that $X \subseteq \cup_{i \in I} U_i$.

Definition 1.2. Let X be a topological space, and let $\{U_i \subset X\}_{i \in I}$ be an open cover of X. Then a *refinement* of this open cover is a set of open subsets $\{V_j \subset X\}_j$ which is still an open cover in itself and such that for each $j \in J$ there exists an $i \in I$ with $V_j \subset U_i$.

Definition 1.3. A topological space X is called *paracompact* if every open cover of X has a refinement by a locally finite open cover.

Note that every compact space is a paracompact.

Definition 1.4. An *abstract simplicial complex* is a collection K of finite subsets of a set X satisfying the following conditions:

(a) For all $x \in X$, $\{x\} \in K$ (i.e., the elements of X belong to K).

(b) If $\tau \in K$, and $\sigma \subseteq \tau$, then $\sigma \in K$.

The elements of K are the *simplices*, and the elements of X are called *vertices* of K.

Next, we state one of the most important concepts to understand how Mapper works. It will be pretty simple to assimilate the mechanics of the algorithm if we get a complete understanding of the nerve concept. The definition is as follows:

Definition 1.5. Let X be a topological space, and $\mathcal{U} = \{U_i\}_{i \in I}$ an open cover of X. The *nerve* of the cover is the abstract simplicial complex $\mathcal{N}(\mathcal{U})$ whose vertex set is $\mathcal{U}$ and satisfies

$$\sigma = [U_{i_0}, U_{i_1}, \ldots, U_{i_k}] \in \mathcal{N}(\mathcal{U}) \iff \cap_{j=0}^{k} U_{i_j} \neq \emptyset.$$

In other words, we can describe the nerve of an open cover as the abstract simplicial complex obtained from the subsets of the proper cover. The vertices of the abstract simplicial complex are the open subsets from the open cover, and the edges between them are created if and only if the intersection of their respective subsets is not empty.

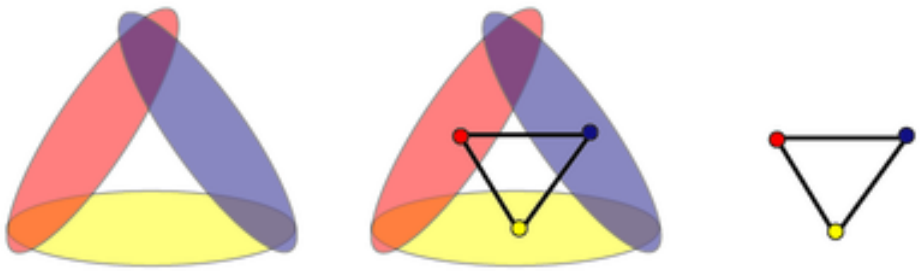

Figure 1.1: *Nerve of an open cover made out of its subsets in the plane.*

Now, we are going to state the necessary homotopical concepts to understand and prove the homotopical version of the Nerve Theorem.

Definition 1.6. Let X and Y be topological spaces. Given two maps $f_0 : X \to Z$, $f_1 : X \to Z$, then f_0 and f_1 are *homotopic* if there exists a continuous map $H : [0,1] \times X \to Z$ satisfying for all $x \in X$:

(a) $H(0,x) = f_0(x)$;

(b) $H(1,x) = f_1(x)$.

If f_0 and f_1 are homotopic, we denote it by $f_0 \sim f_1$.

Note that any two real functions $f_0, f_1 : \mathbb{R} \to \mathbb{R}$ are homotopic. Simply, we just have to define a continuos map $H : \mathbb{R} \times [0,1] \to \mathbb{R}$ such that $H(x,t) = (1-t) * f_0(x) + t * f_1(x)$.

Definition 1.7. Two topological spaces X and Y are *homotopy equivalent* (or have the *same homotopy type*) if there exist continuous maps $f : X \to Y$ and $g : Y \to X$ such that $g \circ f$ is homotopic to Id_X and $f \circ g$ is homotopic to Id_Y.

Definition 1.8. Given a topological space X, we say that X is *contractible* if it is homotopy equivalent to a one-point space.

Now, we are able to state the main result of this section:

Theorem 1.9. (Nerve Theorem) *If $\mathcal{U}$ is an open cover of a paracompact space X such that every nonempty intersection of finitely many sets in $\mathcal{U}$ is contractible, then X is homotopy equivalent to the nerve $\mathcal{N}(\mathcal{U})$.*

We need a proposition and some previous definitions to prove this theorem. We start by giving some definitions:

Definition 1.10. A space X is called a *cell complex*, or *CW complex*, when it is constructed in the following way:

(a) Start with a discrete set X_0 , whose points are called 0-cells.

(b) Inductively, form the n-skeleton X^n from X^{n-1} by attaching n-cells e^n_α via maps $\varphi_\alpha : S^{n-1} \to X^{n-1}$. This means that X^n is the quotient space of the disjoint union $X^{n-1} \bigsqcup_\alpha D^n_\alpha$ of X^{n-1} with a collection of n-disks D^α_n under the identifications $x \sim \varphi_\alpha(x)$ for $x \in \partial D^n_\alpha$. Thus as a set, $X^n = X^{n-1} \bigsqcup_\alpha e^n_\alpha$ where each e^n_α is an open n-disk.

(c) One can either stop this inductive process at a finite stage, setting $X = X^n$ for some $n < \infty$, or one can continue indefinitely, setting $X = \cup_n X^n$ open (or closed) iff $A \cap X^n$ is open (or closed) in X^n for each n.

Remark 1.11. The *n-skeleton* of a topological space X presented as a CW complex (or a simplicial complex) refers to the subspace X_n that is the union of the cells of X (or simplices of X) of dimensions $m \leq n$.

From now on, we will denote the diagram of space simply by X. Then, we have the following definition:

Remark 1.12. A *diagram of spaces* consists of an oriented graph Γ with a space X_i for each vertex i of Γ and a map $f_{(i,j)} : X_i \to X_j$ for each edge $e_{(i,j)}$ of Γ from a vertex i to a vertex j.

Definition 1.13. Given a diagram of spaces X, we define a space $\sqcup X$ to be the quotient of the disjoint union of all the spaces X_v associated to vertices of the graph Γ under the identifications $x \sim f_e(x)$ for all maps f_e associated to edges of Γ. To give a name to this construction, let us call $\sqcup X$ the *amalgamation* of the diagram X.

To get a notion with nicer homotopy-theoretic properties, we introduce the homotopy version of $\sqcup X$, which we shall denote ΔX and call the *realization* of X . Here we again start with the disjoint union of all the vertex spaces X_v, but instead of passing to a quotient space of this disjoint union, we enlarge it by filling in a mapping cylinder M_f for each map f of the diagram, identifying the two ends of this cylinder with the appropriate X_v's.

Remark 1.14. The *mapping cylinder* of a continuous function f between topological spaces X and Y is the quotient $M_f = (([0,1] \times X) \sqcup Y)/\sim$, where $\sim$ is the equivalence relation generated by $(0, x) \sim f(x)$ for each $x \in X$.

Example 1.15. Consider a diagram X of the form $X_0 \leftarrow X_0 \times X_1 \rightarrow X_1$ whose maps are the projections onto the two factors. Since $(a, b) \sim a$ for all $a \in X_0$ and $(a, b) \sim b$ for all $b \in X_1$, we have that $\sqcup X$ is simply a point.

On the other hand, the union of the two mapping cylinders is the same as the quotient of $X_0 \times X_1 \times I$ with $X_0 \times X_1 \times \{0\}$ collapsed to X_0 and $X_0 \times X_1 \times \{1\}$ collapsed to X_1 (i.e., the join $X_0 * X_1$).

Definition 1.16. There is a natural generalization of ΔX in which one starts with a Δ-complex Γ and a diagram of spaces associated to the 1-skeleton of Γ such that the maps corresponding to the edges of each n-simplex of Γ, $n > 1$, form a commutative diagram. We call this data a *complex of spaces*.

If X is a complex of spaces, then for each n-simplex of Γ we have a sequence of maps $X_0 \xrightarrow{f_1} X_1 \xrightarrow{f_2} \dots \xrightarrow{f_n} X_n$, and we define the iterated mapping cylinder $M(f_1, \dots, f_n)$ to be the usual mapping cylinder for $n = 1$, and inductively for $n > 1$, the mapping cylinder of the f_n composition $M(f_1, \dots, f_{n-1}) \rightarrow X_{n-1} \xrightarrow{f_n} X_n$ where the first map is the canonical projection of a mapping cylinder onto its target end.

Proposition 1.17. *When $X_{\mathcal{U}}$ is the complex of spaces associated to an open cover $\mathcal{U} = \{X_i\}$ of a paracompact space X, the map $p : \Delta X_{\mathcal{U}} \rightarrow \sqcup X_{\mathcal{U}} = X$ is a homotopy equivalence.*

Proof. The realization $\Delta X_{\mathcal{U}}$ can also be described as the quotient space of the disjoint union of all the products $X_{i_0} \cap \cdots \cap X_{i_n} \times \Delta^n$, as the subscripts range over sets of $n+1$ distinct indices and $n \geq 0$, with the identifications over the faces of Δ^n using inclusions $X_{i_0} \cap \cdots \cap X_{i_n} \hookrightarrow X_{i_0} \cap \cdots \cap \hat{X}_{i_j} \cap \cdots \cap X_{i_n}$. From this viewpoint, points of $\Delta X_{\mathcal{U}}$ in a given 'fiber' $p^{-1}(x)$ can be written as finite linear combinations $\sum_i t_i x_i$ where $\sum_i t_i = 1$ and x_i is x regarded as a point of X_i, for those X_i's that contain x.

Since X is paracompact there is a partition of unity subordinate to the cover $\mathcal{U}$. This is a family of maps $\varphi_\alpha : X \rightarrow [0, 1]$ satisfying three conditions: The support of each φ_α is contained in some $X_{i(\alpha)}$, only finitely many φ_α's are nonzero near each point of X, and $\sum_\alpha \varphi_\alpha = 1$. Define a section $s : X \rightarrow \Delta X_{\mathcal{U}}$ of p by setting $s(x) = \sum_\alpha \varphi_\alpha(x) x_{i(\alpha)}$.

The figure shows the case $X = S^1$ with a cover by two arcs, the heavy line indicating the image of s. In the general case the section s embeds X as a retract of $\Delta X_{\mathcal{U}}$, and it is a deformation retract since points in fibers $p^{-1}(x)$ can move linearly along line segments to $s(x)$. □

Therefore, a proof of the Nerve Theorem is as follows:

Proof. The Proposition gives a homotopy equivalence $X \simeq \Delta X_{\mathcal{U}}$. Since the nonempty finite intersections of sets in $\mathcal{U}$ are contractible, the earlier proposition implies that the map $\Delta X_{\mathcal{U}} \to \Gamma$ induced by sending each intersection to a point is a homotopy equivalence. Since Γ is the barycentric subdivision of $\mathcal{N}(\mathcal{U})$, the result follows. □

1.2 Homological Nerve Theorem

A brief theoretical framework to understand the homological nerve theorem is given. However, we motivate the reader to check [3] to get a fully detailed understanding about homology.

On the other hand, these theoretical concepts will also be useful in the discussion about the Mapper stability in Chapter 3. In particular, we need them for an understanding of extended persistence.

Definition 1.18. We define a *chain complex* as a sequence of homomorphisms of abelian groups C_i

$$\cdots \to C_{n+1} \xrightarrow{\partial_{n+1}} C_n \xrightarrow{\partial_n} C_{n-1} \to \dots C_1 \xrightarrow{\partial_1} C_0 \xrightarrow{\partial_0} 0$$

where $\partial_n \partial_{n+1} = 0$ for each n.

From $\partial_n \partial_{n+1} = 0$, since $\partial_n \partial_{n+1}(x) = \partial_n(\partial_{n+1}(x))$, it follows that $Im(\partial_{n+1}) \subset Ker(\partial_n)$. Then we have the following:

Definition 1.19. We can define the n^{th} *homology group* of the chain complex to be the quotient group $H_n = Ker(\partial_n)/Im(\partial_{n+1})$. Elements of $Ker(\partial_n)$ are called *cycles* and elements of $Im(\partial_{n+1})$ are *boundaries*.

Moreover, there is an extension of homology groups by augmenting the chain complex with $\mathbb{Z}$, called the reduced homology groups. The definition is as follows:

Definition 1.20. Let us consider a space X and let $C_n(X)$ be the free abelian group with basis the set of singular n-simplices in X. Then, we define the *reduced homology groups* $\tilde{H}_n(X)$ as the homology groups of the augmented chain complex

$$\cdots \to C_2(X) \xrightarrow{\partial_2} C_1(X) \xrightarrow{\partial_1} C_0(X) \xrightarrow{\epsilon} \mathbb{Z} \to 0$$

where $\epsilon(\sum_i n_i \sigma_i) = \sum_i n_i$. Here we had better require X to be nonempty, to avoid having a nontrivial homology group in dimension -1. Since $\epsilon \partial_1 = 0$, ϵ vanishes on Im ∂_1 and hence induces a map $H_0(X) \to \mathbb{Z}$ with kernel $\tilde{H}_0(X)$, so $H_0(X) \approx \tilde{H}_0(X) \oplus \mathbb{Z}$. Obviously, $H_n(X) \approx \tilde{H}_n(X)$ for $n > 0$.

Now, that we have seen all the necessary theoretical concepts, we can state the homological nerve theorem.

For a finite simplicial complex X, let $H_j(X)$ denote the j-th simplicial homology of X with coefficients in some fixed field $\mathbb{F}$. The j-dimensional skeleton of X is denoted by $X^{(j)}$. Let $\mathcal{U} = \{U_i\}_{i \in I}$ be a finite family of subcomplexes of X such that $\cup_{i \in I} U_i = X$. For $\sigma \subset I$ let $U_\sigma = \cap_{i \in I} U_i$. The nerve of $\mathcal{U}$ is the simplicial complex $N = \mathcal{N}(\mathcal{U})$ on the vertex set I whose simplices are all $\sigma \subset I$ such that $U_\sigma \neq \emptyset$. Then,

Theorem 1.21. *If $\tilde{H}_j(U_\sigma) = 0$ for all $\sigma \in N^{(k)}$ and $0 \leq j \leq k - \dim(\sigma)$, then*

(a) $\tilde{H}_j(X) \cong \tilde{H}_j(N)$ for $0 \leq j \leq k$,

(b) If $H_{k+1}(N) \neq 0$ then $H_{k+1}(X) \neq 0$.

To conclude this first chapter, we recap and analyse the obtained results. Note that, under some hypotheses, the Nerve Theorem states that the nerve of a cover of a paracompact topological space X is homotopy equivalent to X. Roughly speaking, we can say that the topological features of an abstract simplicial complex (the nerve of a cover) are related to the topological features of the original topological space.

The Nerve Theorem plays a fundamental role in TDA since it claims solid theoretical guarantees about the topological features of the nerve. Indeed, it provides a way to encode the topology of spaces into abstract combinatorial structures that are well-suited for designing effective data structures and algorithms.

Chapter 2

Principal Components Analysis

To get a better comprehension about the dataset in our study we apply some statistical methods. Some of those tools are really basic, but one of them needs theoretical background to understand the results we obtain. This method is called *Principal Components Analysis*, commonly abbreviated as *PCA*.

PCA is usually used to analyse high dimensional data since it is a great dimensionality reduction method. Even though it reduces the dimensionality of our data it keeps most of the variation of the data set —that is the reason for being one of the most used methods to treat high dimensional data.

Before explaining the methodology of *PCA*, we have to explain some previous theoretical concepts. Firstly, we see some theory for *singular values* and a method that uses them to also reduce the dimension of a point cloud. The method in particular is called *Singular Value Decomposition*. Then, we study the idea behind *principal components*, and, finally, the *Principal Components Analysis*.

2.1 Singular Value Decomposition

Let A be an $m \times n$ matrix, so we have the symmetric matrix A^TA. Denoting $B = A^TA$, we know that M has n eigenvalues and n linearly independent and orthogonal eigenvectors $v_1, v_2, \ldots, v_n$. Assuming λ_i is an eigenvalue and v_i its respective eigenvector, we have that

$$||Bv_i||_2 = (Bv_i)^T Bv_i = v_i^T B^T Bv_i = v_i^T \lambda v_i = \lambda v_i^T v_i = \lambda ||v_i||_2.$$

Thus, since $||Bv_i||_2 \geq 0$, we obtain that $\lambda||v_i||_2 \geq 0$, and, in particular, $\lambda \geq 0$. Hence, since we can do it for any eigenvalue from B, we have proved the following result:

Proposition 2.1. *Let B be a real $n \times n$ matrix, with rank t. Then, every eigenvalue λ of B is positive, i.e., $\lambda \geq 0$.*

From this proposition we get the definition of a *singular value*:

Definition 2.2. Let A be an $m \times n$ matrix, and B the matrix $B = A^T A$. Let $\lambda_1, \lambda_2, \ldots, \lambda_n$ be the n eigenvalues from B ordered in decreasing order, i.e., $\lambda_1 \geq \lambda_2 \geq \cdots \geq \lambda_n$. We define the *n singular values* of A by

$$\sigma_i = \sqrt{\lambda_i}, \text{ for each } i = 1, \ldots, n \text{ such that } \sigma_1 \geq \sigma_2 \geq \cdots \geq \sigma_n.$$

Now that we have stated the definition of singular values, we can introduce the *Singular Value Decomposition*.

Definition 2.3. Given $A \in \mathbb{C}^{m \times n}$ a *Singular Value Decomposition* (*SVD*) of A is a factorization

$$A = U \Sigma V^*,$$

where $U \in \mathbb{C}^{m \times m}$ and $V \in \mathbb{C}^{n \times n}$ are unitary, and $\Sigma \in \mathbb{R}^{m \times n}$ is diagonal.

In addition, it is assumed that the diagonal entries σ_j of Σ satisfy $\sigma_1 \geq \sigma_2 \geq \cdots \geq \sigma_p \geq 0$, where $p = \min(m, n)$. Note that the diagonal matrix Σ has the same shape as A even when A is not square, but U and V are always square unitary matrices.

Note that if $A \in \mathbb{R}^{m \times n}$ then U and V can also be guaranteed to be real orthogonal matrices. Therefore, the *SVD* is commonly denoted by $U \Sigma V^T$. *SVD* is the most commonly used method for data science since, in general, the values in the data sets are real.

Hence, we can see that *SVD* projects our initial data, from a matrix A with r columns to a subspace with r or fewer columns. However, the essence of the original data is conserved. By the previous reason, *SVD* is sometimes a dimensionality reduction method, even though it is also used for other purposes.

2.2 Principal Components Analysis

The *Principal Components Analysis* (*PCA*) method uses the theory behind *principal components* (*PC*). Thus, we need to study these theoretical concepts previously.

2.2.1 Definition of Principal Components

Given a vector of p random variables $x = (x_1, x_2, \ldots, x_p)$, PCA focus on the variance of the random variables[1]. Firstly, we have to find a linear function $\alpha_1^T x$ of

[1] Even though it does not completely ignore their covariances and correlations.

the elements of x having maximum variance, where α_1 is a vector of p constants $\alpha_{1_1}, \alpha_{1_2}, \ldots, \alpha_{1_p}$, so that

$$\alpha_1^T x = \alpha_{1_1} x + \alpha_{1_2} x_2 + \cdots + \alpha_{1_p} x_p = \sum_{j=1}^{p} \alpha_{1_j} x_j.$$

Then, the following step is to find a linear function $\alpha_2^T x$ that is uncorrelated with the previous one, $\alpha_1^T x$, and that has maximum variance. Hence, we keep doing this procedure such that at the kth step, we find a linear function $\alpha_k^T x$ such that is uncorrelated with $\alpha_1^T x, \alpha_2^T x, \ldots, \alpha_{k-1}^T x$ and has a maximum variance. We define the kth variable derived from the procedure described above ($\alpha_k^T x$) as the *kth Principal Component (PC)*.

In general, most of the variation in x will be accounted for by the m first principal components, for $m << p$. Consequently, we do not usually compute all the principal components until the pth one. Notice that this is a pretty relevant and valuable fact to consider in practice.

When a set of $n > 2$ variables has substantial correlations among them, then we will get most of the variation from the original variables in the first PCs. On the other hand, the last few principal components identify directions in which there is a minimal variation; they identify near-constant linear relationships among the original variables.

Obtaining principal components

Now that we have already stated the theory for PCs, we have to define a method to find them. Consider the case where x, the vector of p random variables, has a known covariance matrix Σ. The covariance matrix is the matrix defined as follows:

$$\Sigma = \begin{cases} \text{var}(x_j) & \text{if } i = j \\ \text{cov}(x_i, x_j) & \text{if } i \neq j \end{cases}$$

When Σ is unknown —the most realistic case— we replace Σ with a sample covariance matrix S.

To derive the form of the PCs, consider first $\alpha_1^T x$; the vector α_1 maximizes $\text{var}[\alpha_1^T x] = \alpha_1^T \Sigma \alpha_1$. It is clear that, as it stands, the maximum will not be achieved for finite α_1, so a normalization constraint must be imposed. The constraint used in the derivation is $\alpha_1^T \alpha_1 = 1$, that is, the sum of squares of elements of α_1 equals 1.

To maximise $\alpha_1^T \Sigma \alpha_1$ subject to $\alpha_1^T \alpha_1 = 1$, the standard approach is to use the tech-

nique of Lagrange multipliers. Then, we have to maximise the equation

$$\alpha_1^T \Sigma \alpha_1 - \lambda(\alpha_1^T \alpha_1 - 1),$$

where λ is a Lagrange multiplier. Differentiating the above equation with respect to the variable α_1, we obtain

$$\Sigma\alpha_1 - \lambda\alpha_1 = 0, \text{ that is equivalent to } (\Sigma - \lambda I_p)\alpha_1 = 0,$$

where I_p denotes the identity matrix of dimension p. Thus, λ is an eigenvalue of Σ and α_1 is the corresponding eigenvector. To decide which of the p eigenvectors gives α_1^T with maximum variance, note that the quantity to be maximized is

$$\alpha_1^T \Sigma \alpha_1 = \alpha_1^T \lambda \alpha_1 = \lambda \alpha_1^T \alpha_1 = \lambda,$$

so λ must be as large as possible. Thus, α_1 is the eigenvector corresponding to the largest eigenvalue of Σ, and $var(\alpha_1^T x) = \alpha_1^T \Sigma \alpha_1 = \lambda_1$, the largest eigenvalue. In general, the kth PC of x is $\alpha_k^T x$ and $var(\alpha_k^T x) = \lambda_k$, where λ_k is the kth largest eigenvalue of Σ, and α_k is the corresponding eigenvector[2].

Principal Components for a given sample

Now, we will study the methodology of PC for a given sample. This is equivalent to our case where a given sample has been given to us and we have to obtain the principal components to be our filter functions.

Suppose that we have n independent observations on the p-element random vector x; denote these n observations by $x_1, x_2, \ldots, x_n$. Let $\tilde{z}_{i_1} = a_1^T x_i$, $i = 1, 2, \ldots, n$, and choose the vector of coefficients a_1^T to maximize the sample variance

$$\frac{1}{n-1} \sum_{i=1}^{n} (\tilde{z}_{i_1} - \bar{z}_1)^2$$

subject to the normalization constraint $a_1^T a_1 = 1$. Next let $\tilde{z}_{i_2} = a_2^T x_i$, $i = 1, 2, \ldots, n$, and choose a_2^T to maximize the sample variance of the $\tilde{z}_{i_2}$ subject to the normalization constraint $a_2^T a_2 = 1$, and subject also to the $\tilde{z}_{i_2}$ being uncorrelated with the $\tilde{z}_{i_1}$ in the sample. Proceeding with the same pattern we defined previously in the general case, we get a sample version of the definition of principal components. Then, we can assure that $a_k^T x$ is defined as the kth sample PC, $k = 1, 2, \ldots, p$. Moreover, we introduce a new term, called *score*, such that $\tilde{z}_{i_k}$ is the score for the ith observation on the kth PC. Basically, the score is somehow the "new coordinate" value.

[2]For a detailed proof of this result, check [6]

Following the same derivation strategy seen in the general case, but now with sample variances and covariances replacing population quantities, then it turns out that the sample variance of the PC scores for the kth sample PC is l_k, the kth largest eigenvalue of the sample covariance matrix S for $x_1, x_2, \ldots, x_n$, and a_k is the corresponding eigenvector for $k = 1, 2, \ldots, p$.

Define the $n \times p$ matrices $\tilde{X}$ and $\tilde{Z}$ so that the (i,k)th elements equal to the value of the kth element $\tilde{x}_{i_k}$ of x_i, and to $\tilde{z}_{i_k}$, respectively. Then, $\tilde{Z}$ and $\tilde{X}$ are related by $\tilde{Z} = \tilde{X}A$, where A is the $p \times p$ orthogonal matrix whose kth column is a_k.

If the mean of each element of x is known to be zero, then $S = \frac{1}{n}\tilde{X}^T\tilde{X}$. However, the most common case is when $\bar{x}$ (mean of x) is unknown. Therefore, in this case, we have that the (j,k)th element of S is

$$\frac{1}{n-1}\sum_{i=1}^{n}(\tilde{x}_{i_j} - \bar{x}_j)(\tilde{x}_{i_k} - \bar{x}_k),$$

where $\bar{x}_j = \frac{1}{n}\sum_{i=1}^{n}\tilde{x}_{ij}$, $j = 1, 2, \ldots, p$.

Consequently, the matrix S can be written as follows:

$$S = \frac{1}{n-1}X^T X, \tag{2.1}$$

where X is an $n \times p$ matrix with (i,j)th element $(\tilde{x}_{i_j} - \bar{x}_j)$. The notation x_{i_j} will be used to denote the (i,j)th element of X, so that x_{i_j} is the value of the jth variable *measured about its mean* $\bar{x}_j$ for the ith observation.

Moreover, it can also be more convenient to define the matrix of PC scores as

$$Z = XA. \tag{2.2}$$

These PC scores will have the same variances and covariances as those given by $\tilde{Z}$, although these ones will have zero means, rather than means $\bar{z}_k$, $k = 1, 2, \ldots, p$.

Notice that given the X^TX eigenvalues $\lambda_1, \ldots, \lambda_p$ we have that the matrix $C = \frac{1}{n-1}X^TX$ will have the same eigenvectors $\lambda_1, \ldots, \lambda_p$. Furthermore, for $v_1, \ldots, v_p$, i.e., eigenvectors of X^TX, we know that the eigenvectors of C are defined by $\frac{1}{n-1}v_i$, for $i = 1, \ldots, p$. Hence, this implies that sometimes it can be more convenient to work in terms of eigenvalues and eigenvectors of X^TX, rather than directly with those of S.

2.2.2 Principal Components Analysis

Intuitively, Principal Components Analysis is a statistical procedure or method that reduces the dimensionality of a dataset with many variables while preserving most of the information given by the initial data. In particular, PCA retains most of the variation from the original dataset. This dimensionality reduction is made through the earlier defined principal components since they become a new set of variables with most of the original information. Remember that the first few PCs retain most of the variation present in initial variables and are uncorrelated between them. Then, we have to compute the principal components that reduce to some eigenvalue-eigenvector problem for a positive-semidefinite symmetric matrix. However, we can also obtain the scores of the initial data and the loadings, i.e., the weight of the original variables into the principal components.

2.3 *SVD & PCA*

SVD provides a computationally efficient method of actually finding PCs. Since we can find U, Σ, V satisfying the *SVD* factorisation, then V and Σ will give us the eigenvectors and the square roots of the eigenvalues of $X^T X$. Consequently, we can obtain the coefficients and standard deviations of the principal components for the sample covariance matrix.

We can also get the scaled versions of the principal components scores from the matrix U. We can see it by multiplying the factorization obtained for *SVD* on the right by V, such that: $XV = U\Sigma V^T V = U\Sigma$, as $V^T V = I_r$. Therefore, since the matrix XV has the PC scores for the kth PC as its kth column, we have that the PC scores z_{i_k} are given by

$$z_{i_k} = u_{i_k} \sigma_k^{1/2}, \;\; i = 1, 2, \ldots, n, \;\; k = 1, 2, \ldots, r,$$

or, equivalently, $U = Z\Sigma^{-1}$. The variance of the scores for the kth PC is $\dfrac{\sigma_k}{(n-1)}$, $k = 1, 2, \ldots, p$. Then, U gives the scores of Z but scaled to have variance $1/(n-1)$.

Moreover, another point concerning the SVD is that it provides simultaneously not only the coefficients and variances for the PCs, but also the scores of each observation on each PC. The PC scores would otherwise need to be derived as an extra step after calculating the eigenvalues and eigenvectors of the covariance or correlation matrix $S = \dfrac{1}{n-1} X^T X$.

Chapter 3

Mapper and its Stability

Mapper is a *TDA* technique that has been used to examine high-data sets. Mapper is an algorithm introduced in 2007 by Singh, Memoli and Carlsson in their seminal paper *Topological Methods for the Analysis of High Dimensional Data Sets and 3D Object Recognition*.

Notice that the algorithm is relatively recent even though Mapper has been applied to different areas (e.g., medicine and sports) for the last years. One of the issues about its newness is that it is still being developed and studied, and some concepts nowadays are not clear yet. An example might be the disinformation about its stability and parameter selection that are left to the user's choice.

Generally, Mapper does not satisfy the Nerve Theorem because many inputs influence the results. Further in this reading, we will see that it is pretty hard when applying the algorithm to satisfy the hypothesis of the Nerve Theorem. Consequently, this somehow justifies the fact that the Nerve Theorem is not generally satisfied when applying Mapper.

However, the Nerve Theorem is an important link between topological spaces and discrete geometric and topological algorithms. It is at the heart, either implicitly or explicitly, of many foundational algorithms in topological data analysis. Hence, we could say that the idea of the Nerve Theorem is behind to creation of the Mapper algorithm.

In this chapter, we will see some results about the stability of one-dimensional Mapper. The stated results are focused on the stability based on the perturbation of the parameters the user has to select before applying the algorithm.

Nevertheless, in the practical part of our study, we have applied the 2-dimensional algorithm. We chose 2-dimensional Mapper because it was more convenient for us, and it gave much better results for the given data. We will give strong reasons

justifying the decision further in this section.

We divide this section into two; the first one has a detailed explanation of how Mapper works and the application of the previously seen theoretical concepts to it. Then, we state the results obtained in the paper mentioned above about the stability in $1D$ Mapper.

3.1 Mapper

The idea of the Mapper algorithm is, given a data set X and a well-chosen real valued function $f : X \to \mathbb{R}^d$, to summarize X through the nerve of the refined pull back of a cover $\mathcal{U}$ of $f(X)$. For well-chosen covers $\mathcal{U}$, this nerve is a graph providing an easy and convenient way to visualize a summary of the data.

Definition 3.1. Let $f : X \to \mathbb{R}^d$, $d \geq 1$, be a continuous real valued function and let $\mathcal{U} = (U_i)_{i \in I}$ be a cover of $\mathbb{R}^d$. The *pull back cover* of X induced by $(f,\mathcal{U})$ is the collection of open sets $(f^{-1}(U_i))_{i \in I}$. The *refined pull back* is the collection of connected components of the open sets $f^{-1}(U_i)$, $i \in I$.

The Mapper Algorithm

Given a data set X, we filter the values of X using a filter function, commonly called *lens*, $f : X \to \mathbb{R}^d$. Then, once we have the filtered set $f(X)$, we get a cover $\mathcal{U} = \bigcup_{i \in I} U_i$ of $f(X)$ and its pull back cover, i.e., the collection of open sets $(f^{-1}(U_i))_{i \in I}$. Thereafter, we decompose $f^{-1}(U_i)$ into clusters $C_{U_i,1}, \ldots, C_{U_i,k_{U_i}}$ for every $i \in I$ using a clustering algorithm. Finally, we introduce the nerve concept we have mentioned earlier, such that we compute the nerve of the cover of $C_{U_i,1}, \ldots, C_{U_i,k_{U_i}}$, $U_i \in \mathcal{U} = \bigcup_{i \in I} U_i$ of X.

Thus, by definition, we obtain an abstract simplicial complex, and ideally, it should be homotopy equivalent to X. However, as we stated before, this is generally not satisfied. The great range of choices for the lens, number of intervals, intervals overlapping percentage, and the clustering algorithm implies that the Nerve Theorem will not always apply for Mapper. In fact, Mapper is not even stable on a big scale, even though it can be stable for minimal changes. Further in this paper, we discuss these topics and give some formal results.

Moreover, it can be noted that the Mapper algorithm has a straightforward structure from the very definition. Nevertheless, it raises several questions about the various choices left to the user. Hence, we also try to briefly discuss how to select the most optimal parameters and get some valuable conclusions for the algorithm implementation.

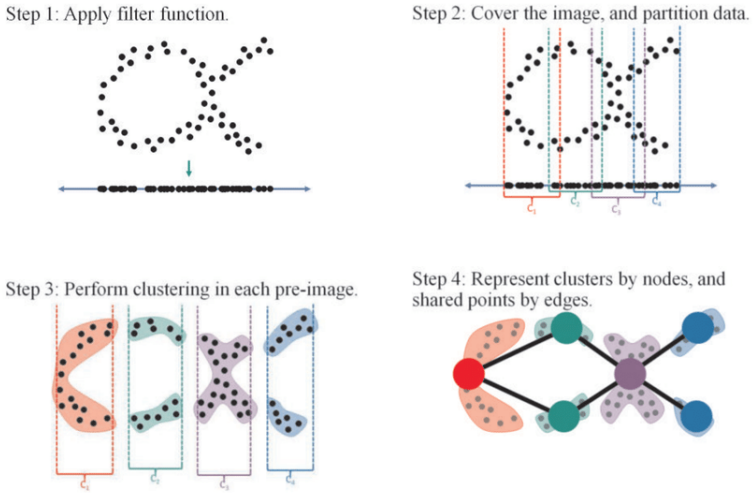

Figure 3.1: Visual representation example of the Mapper algorithm's application for a given dataset.

The choice of f. The choice of the function f, sometimes called the filter function or *lens*, strongly depends on the data features that one expects to highlight. Hence, it is essential to analyse our data thoroughly before choosing the filter function. That is the reason why, in our case, we carry out a *PCA* analysis. The two following lenses are the most used ones:

(a) The eigenfunctions given by a Principal Component Analysis of the data.

(b) The centrality function $f(x) = \sum_{y \in X} d(x,y)$ and the eccentricity function $f(x) = \sup_{y \in X} d(x,y)$ sometimes appear to be good choices that do not require any specific knowledge about the data.

The choice of the cover $\mathcal{U}$. The output of the Mapper is very sensitive to the choice of $\mathcal{U}$, and small changes in the number of intervals and its overlapping percentage parameters may result in substantial changes in the output, making the method very unstable. Note that overlaps will determine the edge creation in the Mapper's graph. A classical strategy consists of exploring a range of parameters and selecting the ones that turn out to provide the most informative output from the user perspective.

The choice of the clusters. The Mapper algorithm requires to cluster the preimage of the open sets $U \in \mathcal{U}$. There are two strategies to compute the clusters. A first strategy consists of applying, for each $U \in \mathcal{U}$, the clustering algorithm (chosen by the user) to the preimage $f^{-1}(U)$. A second and more global strategy consists

in building a neighbouring graph on top of the data set X, e.g., k-NN graph or ϵ-graph, and, for each $U \in \mathcal{U}$, taking the connected components of the subgraph with vertex set $f^{-1}(U)$. Mapper does not place any conditions on the clustering algorithm. Thus any domain-specific clustering algorithm can be used. Some of the most implemented clustering algorithms are:

(a) *K-means* is the most commonly used clustering algorithm. It is a *centroid-based* and unsupervised learning algorithm. The numbers of final clusters are defined by K, so the user can choose the number of clusters the dataset is divided with. First, we assign each point to a cluster randomly. Then, determine the cluster centroid coordinates. Finally, we repeat the following steps until we reach a situation where there is no improvement by switching points from one cluster to another. The steps consist of determining each data point's distances to the centroids, re-assigning each point to the closest cluster centroid based upon minimum distance, and calculating cluster centroids again.

(b) *DBSCAN* is, unlike *K*-means, a density-based clustering algorithm.

In our study we apply the two principal components as our filter function and the *K*-means algorithm. Henceforth, we argue the implementation of the Nerve Theorem into our dataset and the chosen parameters.

3.2 Nerve Theorem into our Mapper

In order to briefly discuss the relation between the Nerve Theorem and the Mapper algorithm in our case, we will examine each point of the Nerve Theorem in our dataset. Henceforth, we study the paracompactness of our space and the intersection of finitely many sets of the open cover $\mathcal{C}$.

We will begin by analysing the paracompact property of our dataset. Note that our point cloud is in $\mathbb{R}^4$. Then, we see a general result for $\mathbb{R}^n$ to check the mentioned above property. However, we need to define some terms and concepts to reach this final theorem.

Definition 3.2. Given a topological space X, we say that X is a *Lindelöf space* if every open cover of X has a countable subcover.

Definition 3.3. A topological space X is *regular* if for any point $x \in X$ and closed set A, $x \notin A$, there exists open sets U_x and V_A such that $x \in U_x$, $A \subset V_A$ and $U_x \cap V_A = \emptyset$.

Now that we have stated all the necessary definitions, we need to state the following theorem:

Theorem 3.4. *All regular Lindelöf spaces are paracompact.*

Hence, since $\mathbb{R}^n$ is a Lindelöf space, we have that $\mathbb{R}^n$ is indeed a paracompact space. Moreover, we now want to argue that our dataset is also a paracompact space. In order to get there, we need the following proposition:

Proposition 3.5. *Every closed subspace of a paracompact space X is paracompact.*

Therefore, we have that points are closed sets in $\mathbb{R}^4$ and that the finite union of closed sets is closed. Since we have a finite number of points, we can state that our point cloud is a closed set. Thus, since it is a closed subspace of $\mathbb{R}^4$, we can conclude that our dataset is a paracompact space.

As we wanted to prove, we have concluded that, in general, i.e., when our dataset is in $\mathbb{R}^d$, for $d \leq 1$, the dataset we analyse is a paracompact space. This fact implies that the paracompactness requirement of the Nerve Theorem is usually satisfied. Then, the validity of the Nerve Theorem over the Mapper algorithm relies on the second hypothesis, i.e., "*every nonempty intersection of finitely many sets in $\mathcal{U}$ is contractible*."

In order to discuss the second hypothesis of the Nerve Theorem, we define a previous concept:

Definition 3.6. Given a topological space X, we say that an open cover $\mathcal{U} = \{U_i\}_{i \in I}$ of X is a good cover if all the U_i's and all their inhabited finite intersections are contractible topological spaces.

Therefore, it is hard to prove that the final open cover of X obtained by the Mapper algorithm is a good cover. This fact relies, among others[1], on the non-stability of the clustering algorithms. In particular, the K-means algorithm does not always give the same results since different slightly different clusters can result from the initialisation of different centroids.

However, even though in our case, and in general, the paracompactness of the initial space is satisfied, we have seen that it is hard to prove the good cover property. Thus, it is hard to guarantee that the Nerve Theorem will always be satisfied for the whole range of parameters are open to being chosen.

[1]There is a wide range of parameter options to choose from that can affect the good cover property. However, we focus on the clustering algorithm since it is the clearest.

The following section of this chapter tries to state some results about the stability of the Mapper under parameters perturbations. We formally quantify the possible variances on the Mapper graphs after changing some of the initial parameters. The possible variances can be translated as the different topological features from one plot to another. The discussion is focused on the 1-dimensional Mapper, even though we apply the 2-dimensional one in our study.

3.3 Structure and Stability of the 1-Dimensional Mapper

As we mentioned earlier, we begin by stating some theoretical concepts that will be needed to understand to examine the relation between the Mapper and the *Reeb graph*. This relation will able us to examine and predict the appearance, or disappearance, of new features given by different filter functions and covers. Moreover, this proposed theoretical framework will facilitate the quantification of the stability of the Mapper structure. We can modify the cover by choosing different intervals, overlapping percentages, and even clustering algorithms. Then, we guarantee some guarantees about the stability between Mapper graphs under small changes in their initial conditions.

The connection between the Mapper and Reeb graph needs an intermediate object called the Multinerve Mapper. Then, we begin by studying the Mapper and Multinerve Mapper relation, using the theory behind the Nerve and Multinerve connection. Afterwards, given a pair (X, f), we determine the relationship between the Multinerve Mapper and Reeb graph.

3.3.1 Extended Persistence

This section introduces the definition of extended persistence and the necessary concepts to get to it. The extended persistence provides a method to relate the Mapper with Reeb graphs. Thus, we first introduce the definition of a Morse type function:

Definition 3.7. Consider a function $f : X \to \mathbb{R}$ on a topological space X. We say that f is a *Morse type* function if:

(a) There exists a finite set $Crit(f) = \{a_1 < \cdots < a_n\}$, called *set of critical values*, such that over all open intervals $(a_0 = -\infty, a1), \ldots, (a_i, a_{i+1}), \ldots, (a_n, a_{n+1} = +\infty)$ there is a compact and locally connected space Y_i and a homeomorphism $\mu_i = Yi \times (a_i, a_{i+1}) \to X^{(a_i, a_{i+1})}$ such that for $i = 0, \ldots, n$, $f_{|X^{(a_i, a_{i+1})}} = \pi_2 \circ \mu_i^{-1}$ (π_2 is the projection onto the second factor).

(b) For $i = 1, \ldots, n-1$, μ_i extends to a continuous function $\hat{\mu}_i : Y_i \times [a_i, a_{i+1}] \to X^{[a_i, a_{i+1}]}$. Similarly, μ_0 extends to $\hat{\mu}_0 : Y_0 \times (-\infty, a_1] \to X^{(-\infty, a_1]}$ and μ_n extends to $\hat{\mu}_n : Y_n \times (a_n, +\infty) \to X^{(a_n, \infty]}$.

(c) Each element X^t has finitely-generated homology.

Definition 3.8. A *filtration* $\mathcal{F}$ is an indexed family $(S_i)_{i \in I}$ of subobjects of a given algebraic structure S, with the index i running over some totally ordered index set I, subject to the condition that if $i \leq j$ in I then $S_i \subseteq S_j$.

Proposition 3.9. *Consider the function $f : X \to \mathbb{R}$, where X is a topological space. The family $\{X^{(-\infty,\infty)}\}_{\alpha \in \mathbb{R}}$ of sublevel sets of f defines a filtration, i.e., it is nested with respect to inclusion $X^{(-\infty,\alpha]} \subseteq X^{(-\infty,\beta]}$ for all $\alpha \leq \beta \in \mathbb{R}$.*

The family $\{X^{[\alpha,+\infty)}\}_{\alpha \in \mathbb{R}}$ of superlevel sets of f is also nested but in the opposite direction: $X^{[\alpha,+\infty)} \supseteq X^{[\beta,+\infty)}$ for all $\alpha \leq \beta \in \mathbb{R}$. Considering $\mathbb{R}^{op} = \{\tilde{x} : x \in \mathbb{R}\}$, ordered by $\tilde{x} \leq \tilde{y}$ iff $x \geq y$, we can index the family of superlevel sets by $\mathbb{R}^{op}$. Then, we have a filtration: $\{X^{[\tilde{\alpha},+\infty)}\}_{\tilde{\alpha} \in \mathbb{R}^{op}}$, with $X^{[\tilde{\alpha},+\infty)} \subseteq X^{[\tilde{\beta},+\infty)}$ for all $\tilde{\alpha} \leq \tilde{\beta} \in \mathbb{R}^{op}$.

Now, we would like to connect the two filtrations at infinity, and that is what *extended persistence* does. The procedure is the following:

Replace each superlevel set $X^{[\tilde{\alpha},+\infty)}$ by the pair of spaces $(X, X^{[\tilde{\alpha},+\infty)})$. Notice that this maintains the filtration property since we have $(X, X^{[\tilde{\alpha},+\infty)}) \subseteq (X, X^{[\tilde{\beta},+\infty)})$ for all $\tilde{\alpha} \leq \tilde{\beta} \in \mathbb{R}^{op}$. Then, let $\mathbb{R}_{Ext} = \mathbb{R} \cup \{+\infty\} \cup \mathbb{R}^{op}$, where the order is completed by $\alpha < +\infty < \tilde{\beta}$ for all $\alpha \in \mathbb{R}$ and $\tilde{\beta} \in \mathbb{R}^{op}$. This poset (partially ordered set) is isomorphic to $(\mathbb{R}, \leq)$.

Definition 3.10. Finally, we define the *extended filtration* of f over $\mathbb{R}_{Ext}$ by

$$\begin{cases} F_\alpha = X^{(-\infty,\alpha]}, \text{ for } \alpha \in \mathbb{R}, \\ F_{+\infty} = X \equiv (X, \emptyset), \\ F_{\tilde{\alpha}} = (X, X^{[\tilde{\alpha},+\infty)}), \text{ for } \tilde{\alpha} \in \mathbb{R}^{op}. \end{cases}$$

Remark 3.11. This is a well-defined filtration since we have $X^{(-\infty,\alpha]} \subseteq X \equiv (X, \emptyset) \subseteq (X, X^{[\tilde{\beta},+\infty)})$ for all $\alpha \in \mathbb{R}$ and $\tilde{\beta} \in \mathbb{R}^{op}$.

The subfamily $\{F_\alpha\}_{\alpha \in \mathbb{R}}$ is called the *ordinary part* of the filtration, and the subfamily $\{F_{\tilde{\alpha}}\}_{\tilde{\alpha} \in \mathbb{R}^{op}}$ is called the *relative part*.

Formally, singular homology can be regarded as a sequence of functions H_n that assign to each space X an abelian group $H_n(X)$ and to each map $f : X \to Y$ a homomorphism $H_n(f) = f : H_n(X) \to H_n(Y)$, and similarly for relative homology groups. Since these situations are common, we introduce some terminology calling 'functions' like H_n, 'functors', and the domains and ranges of these functors, 'categories'.

Definition 3.12. By applying the homology functor H_* to this filtration, we obtain the so-called *extended persistence module* $EP(f)$:

$$\begin{cases} EP(f)_\alpha = H_*(F_\alpha) = H_*(X^{(-\infty,\alpha]}), \text{ for } \alpha \in \mathbb{R}, \\ EP(f)_{+\infty} = H_*(F_{+\infty}) = H_*(X) \cong H_*(X, \emptyset), \\ EP(f)_{\tilde{\alpha}} = H_*(F_{\tilde{\alpha}}) = H_*(X, X^{[\tilde{\alpha},+\infty)}), \text{ for } \tilde{\alpha} \in \mathbb{R}^{op}, \end{cases}$$

where linear maps between the spaces are induced by the inclusions in the extended filtration.

For *Morse-type* functions, the extended persistence module can be decomposed as a finite direct sum of closed-open *interval modules*

$$EP(f) \simeq \oplus \mathbb{I}_{k=1}^{n}[b_k, d_k],$$

where each summand $\mathbb{I}[b_k, d_k]$ is made of copies of the field of coefficients at each index $\alpha \in [b_k, d_k)$, and copies of the zero space elsewhere, the maps between copies of the field being identities. Each summand represents the lifespan of a homological feature (connected component, hole, void, etc.) within the filtration.

Then, a convenient way to represent the module's structure is to plot each interval in the decomposition as a point in the extended plane, whose coordinates are given by the endpoints. Such a plot is called the *extended persistence diagram* of f, denoted by $Dg(f)$. The distinction between ordinary and relative parts of the filtration allows to classify the points in $Dg(f)$ in the following way:

(a) points whose coordinates both belong to $\mathbb{R}$ are called *ordinary points*; they correspond to homological features being born and then dying in the ordinary part of the filtration;

(b) points whose coordinates both belong to $\mathbb{R}^{op}$ are called *relative points*; they correspond to homological features being born and then dying in the relative part of the filtration;

(c) points whose abscissa belongs to $\mathbb{R}$ and whose ordinate belongs to $\mathbb{R}^{op}$ are called extended points; they correspond to homological features being born in the ordinary part and then dying in the relative part of the filtration.

It is common to decomposose $Dg(f)$ according to this classification:

$$Dg(f) = Ord(f) \sqcup Rel(f) \sqcup Ext^+(f) \sqcup Ext^-(f).$$

Moreover, we give a result, without proof, about the stability of the extended persistence diagrams in d_b^∞. The result is as follows:

Proposition 3.13. *The extended persistence diagrams are stable with respect to the bottleneck distance d_b^∞.*

3.3.2 Reeb Graphs

Let us continue introducing new concepts that we will need later:

Definition 3.2.X. Given a topological space X and a continuous function $f : X \to \mathbb{R}$, we define an equivalence relation $\sim_f$ between points of X by

$$x \sim y \iff [f(x) = f(y), \text{ and } x, y \text{ belong to the same connected component of } f^{-1}(f(x)) = f^{-1}(f(y))].$$

The *Reeb graph* $R_f(x)$ is the space $X/\sim_f$.

There is an interpretation of $Dg(f)$ in terms of the structure of $R_f(X)$. Orienting the Reeb graph vertically so $\tilde{f}$ is the height function, we can see each connected component of the graph as a trunk with multiple branches (oriented upwards, or oriented downwards) and holes. If the *vertical span* of a feature is the span of its image by $\tilde{f}$, we have the following correspondences:

1. The vertical spans of the trunks are given by the points in $Ext_0^+(\tilde{f})$;
2. The vertical spans of the branches that are oriented downwards are given by the points in $Ord_0(\tilde{f})$;
3. The vertical spans of the branches that are oriented upwards are given by the points in $Rel_1(\tilde{f})$;
4. The vertical spans of the holes are given by the points in $Ext_1(\tilde{f})$.

These correspondences provide a dictionary to read off the structure of the Reeb graph from the extended persistence diagram of the induced map $\tilde{f}$. Note that it is a bag-of-features type signature, taking an inventory of all the features (trunks, branches, holes) together with their vertical spans, but leaving aside the actual layout of the features. As a consequence, it is an incomplete signature: two Reeb graphs with the same persistence diagram may not be isomorphic.

3.3.3 MultiNerve Mapper

Definition 3.14. A *simplicial poset* is a partially ordered set $(P, \preceq)$, whose elements are called simplices, and which satisfies the two following properties:

(a) P has a least element called 0 such that $0 \preceq p$ for all $p \in P$.

(b) For all $p \in P$ there exists $d \in \mathbb{N}$ such that the lower segment $[0, p] = \{q \in P : q \preceq p\}$ is isomorphic to the set of simplices of the standard d-simplex with the inclusion as a partial order, where an isomorphism between posets is a bijective and order-preserving function.

From now on, all covers of $Z \subseteq \mathbb{R}$ will be generic, open, minimal, interval covers (*gomic* for short).

Definition 3.15. Let $\mathcal{U} = \{U_\alpha\}_{\alpha \in A}$ be a cover of topological space X. The *multinerve* $\mathcal{M}(\mathcal{U})$ is the simplicial poset defined by

$$\mathcal{M}(\mathcal{U}) := \{(\{\alpha_0, \ldots, \alpha_k\}, C) := \cap_{i=0}^{k} U_{\alpha_i})\} \neq \emptyset \\ \text{and C is a connected component of} \cap_{i=0}^{k} U_{\alpha_i})\}.$$

Given a connected pullback cover $\mathcal{V}$, we extend the Mapper by using the multinerve $\mathcal{M}(\mathcal{V})$ instead of $\mathcal{N}(\mathcal{V})$. This variant will be referred to as the *MultiNerve Mapper* in the following.

Definition 3.16. Let X, Z be topological spaces, $f : X \to Z$ be a continuous function, $\mathcal{U}$ be a cover of $im(f)$ and $\mathcal{V}$ be the associated connected pullback cover. The *Multinerve Mapper* of X is $\bar{\mathcal{M}}_f(X, \mathcal{U}) = \mathcal{M}(\mathcal{V})$.

The connection between the Mapper and the MultiNerve Mapper is induced by the following connection between nerves and multinerves:

Lemma 3.17. *Let X be a topological space and $\mathcal{U}$ a cover of X. Let $\pi_1 : (F, C) \mapsto F$ be the projection of the simplices of $\mathcal{M}(\mathcal{U})$ onto the first coordinate. Then, $\pi_1(\mathcal{M}(\mathcal{U})) = \mathcal{N}(\mathcal{U})$.*

Corollary 3.18. *Let X, Z be topological spaces and $f : X \to Z$ continuous. Let $\mathcal{U}$ be a cover of $im(f)$. Then, $M_f(X, \mathcal{U}) = \pi_1(\bar{M}_f(X, \mathcal{U}))$.*

Lemma 3.19. *When $Z = \mathbb{R}$ and $\mathcal{U}$ is a gomic, π_1 induces a surjective homomorphism in homology.*

The MultiNerve Mapper can be read off from the extended persistence diagram of the Reeb graph. In particular, the MultiNerve Mapper $\bar{M}_f(X,\mathcal{I})$ is actually isomorphic (as a combinatorial multigraph) to a specific Reeb graph, whose extended persistence diagram is realted to the extended persistence diagram $Dg(\tilde{f})$ of $R_f(X)$.[2].

Theorem 3.20. *Let X be a topological space and $f : X \to \mathbb{R}$ be a Morse-type function. Let $R_f(X)$ be the corresponding Reeb graph and $\tilde{f} : R_f(X) \to \mathbb{R}$ be the induced map. Let $\mathcal{I}$ be a gomic of $im(f)$. There are bijections between:*

$$(i)\ Ord_0(\bar{m}_{\mathcal{I}}) \text{ and } Ord_0(\tilde{f}) \setminus Q_O^{\mathcal{I}},\quad (iii)\ Ext_1^-(\bar{m}_{\rangle}) \text{ and } Ext_1^-(\tilde{f}) \setminus Q_{E^-}^{\mathcal{I}},$$
$$(ii)\ Rel_1(\bar{m}_1) \text{ and } Rel_1(\tilde{f}) \setminus Q_R^{\mathcal{I}} \qquad (iv)\ Ext_0^+(\bar{m}_{\mathcal{I}}) \text{ and } Ext_0^+(\tilde{f}).$$

where $Q_O^{\mathcal{I}} = \cup_{I\in\mathcal{I}} Q^+_{\tilde{I}\cup I^+_\cap}$, $Q_R^{\mathcal{I}} = \cup_{I\in\mathcal{I}} Q^-_{\tilde{I}\cup I^+_\cap}$, and $Q_{E^-}^{\mathcal{I}} = \cup_{I\in\mathcal{I}} Q^-_I$, and where, for any interval I with endpoints $a \le b$, we let $Q_I^+ = \{(x,y) \in \mathbb{R}^2 : a \le x \le y \le b\}$ be the half-square strictly below the diagonal.

Using the theorem above we have that the topological features of $\bar{\mathcal{M}}_f(X,\mathcal{I})$ are in bijection with the points of $Dg(\tilde{f})$ minus the ones that fall into the various staircases $(\mathcal{Q}_O^{\mathcal{I}}, \mathcal{Q}_{E^-}^{\mathcal{I}}, \mathcal{Q}_R^{\mathcal{I}})$ corresponding to their type. Where $Dg(\tilde{f})$ is the persistence diagram of the induced function $\tilde{f} : R_f(X) \to \mathbb{R}$, s.t. $f = \tilde{f} \circ \pi$

Moreover, $Dg(\tilde{f})$ itself is obtained from $Dg_0(f)$ and $Dg_1(f)$ by removing the points of $Ext_1^+(f)$ and $Ord_1(f)$. Hence, we use the off-staircase part of $Dg(\tilde{f})$ as a signature for the structure of the Multinerve Mapper.

$$Dg(\bar{\mathcal{M}}_f(X,\mathcal{I})) = (Ord_0(f) \setminus \mathcal{Q}_0^{\mathcal{I}}) \cup ((Ext_0^+(f) \cup Ext_1^-) \setminus \mathcal{Q}^{\mathcal{I}_{E^-}}) \cup (Rel_1(f) \setminus \mathcal{Q}^{\mathcal{I}}). \tag{3.1}$$

We call this signature the *extended persistence diagram of the MultiNerve Mapper*. The fact that $Dg(\bar{M}_f(X,\mathcal{I})) \subseteq Dg(\tilde{f})$ formalizes the intuition that the MultiNerve Mapper should be viewed as a *pixelized version* of the Reeb graph, in which some of the features disappear due to the staircases (prescribed by the cover).

3.3.4 Stability in the bottleneck distance

Given a point in the signature $Dg(\bar{M}_f(X,\mathcal{I}))$, we will calculate the l^∞-distance to its staircase to quantify the needed perturbation of f, or $\mathcal{I}$, to eliminate the corresponding feature in the MultiNerve Mapper. On the other hand, to know

[2]For further details, check [8]

the necessary perturbation to do over f, or $\mathcal{I}$, in order to create a feature in the MultiNerve Mapper is given by the l^∞-distance to the boundary of the staircase of a point that is in the Reeb graph's signature $Dg(\tilde{f})$, but not in the MultiNerve Mapper's.

Now, we define some concepts to be able to define the *bottleneck distance* later. Then, once we get the *bottleneck distance* definition, we will extend it to obtain a distance between extended persistence diagrams. The concepts are the following:

Definition 3.21. Given two persistence diagrams D, D', a *partial matching* between D and D' is a subset Γ of $D \times D'$ such that:

$$\forall p \in D, \text{ there is at most one } p' \in D' \text{ s.t. } (p, p') \in \Gamma$$
$$\forall p' \in D', \text{ there is at most one } p \in D \text{ s.t. } (p, p') \in \Gamma$$

Furthermore, Γ must match points of the same type (ordinary, relative, extended) and of the same homological dimension only.

Definition 3.22. Let Θ be a subset of $\mathbb{R}^2$. Given a partial matching Γ between two extended persistence diagrams $Dg, Dg\prime$, the Θ-*cost* of Γ is:

$$cost_\Theta(\Gamma) = max\{\max_{p \in D_g} \delta_{D_g}(p), \max_{p\prime \in D_{g\prime}} \delta_{D_{g\prime}(p\prime)}\}$$

where:

$$\delta_{Dg}(p) = ||p - p\prime||_\infty \text{ if } \exists p\prime \in Dg\prime \text{ s.t. } (p, p\prime) \in \Gamma \text{ and } d_\infty(p, \Theta) \text{ otherwise,}$$

$$\delta_{Dg\prime}(p\prime) = ||p - p\prime||_\infty \text{ if } \exists p \in Dg \text{ s.t. } (p, p\prime) \in \Gamma \text{ and } d_\infty(p\prime, \Theta) \text{ otherwise,}$$

Definition 3.23. The *bottleneck distance* becomes:

$$d_{b,\Theta}(D_g, D_{g\prime}) = \inf_\Gamma cost_\Theta(\Gamma),$$

where Γ ranges over all partial matchings between Dg and $Dg\prime$.

Thus, we define the distance between signatures as follows:

Definition 3.24. Given a gomic $\mathcal{I}$, we define the distance $d_\mathcal{I}$ between extended persistence diagrams $Dg, Dg\prime$ as:

$$d_\mathcal{I}(Dg, Dg\prime) = \max\{d_{b,\mathcal{Q}^\mathcal{I}_{E^-}}(Ext, Ext\prime), d_{b,\mathcal{Q}^\mathcal{I}_O}(Ord, Ord\prime), d_{b,\mathcal{Q}^\mathcal{I}_R}(Rel, Rel\prime)\}$$

Stability w.r.t Perturbations of the Function

The distance $d_{\mathcal{I}}$ stabilizes the (MultiNerve) Mappers, as stated in the following theorem:

Theorem 3.25. *Given a topological space X, Morse-type functions $f, g : X \to \mathbb{R}$ and a gomic $\mathcal{I}$ of granularity at most $\epsilon > 0$, the following stability inequality holds:*

$$\begin{aligned} d_{\mathcal{I}}(Dg(M_f(X,\mathcal{I})), Dg(M_g(X,\mathcal{I}))) &\leq d_{\mathcal{I}}(Dg(\bar{M}_f(X,\mathcal{I})), Dg(\bar{M}_g(X,\mathcal{I}))) \\ &\leq ||f-g||_\infty. \end{aligned}$$

Moreover, $d_{\mathcal{I}}$ and d_b are related as follows:

$$\begin{aligned} d_b(Dg(\bar{M}_f(X,\mathcal{I})), Dg(\bar{M}_g(X,\mathcal{I}))) &\leq \frac{\epsilon}{2} + d_{\mathcal{I}}(Dg(\bar{M}_f(X,\mathcal{I})), Dg(\bar{M}_g(X,\mathcal{I}))) \\ d_b(Dg(M_f(X,\mathcal{I})), Dg(M_g(X,\mathcal{I}))) &\leq \epsilon + d_{\mathcal{I}}(Dg(M_f(X,\mathcal{I})), Dg(M_g(X,\mathcal{I}))) \end{aligned}$$

Interpretation of the Stability. Denoting by $Q_p^{\mathcal{I}}$ the staircase corresponding to the type of a diagram point p, the quantity

$$d_{\mathcal{I}}(Dg, \emptyset) = \max_{p \in Dg} d_\infty(p, Q_p^{\mathcal{I}})$$

measures the amount by which the diagram Dg must be perturbed in the metric $d_{\mathcal{I}}$ in order to bring all its points to the staircase. Hence, by Theorem 1.2.10., given a pair (X, f), the quantity

$$d_{\mathcal{I}}(Dg(\bar{M}_f(X,\mathcal{I})), \emptyset) = \max_{p \in Dg(\bar{M}_f(X,\mathcal{I}))} d_\infty(p, Q_p^{\mathcal{I}})$$

is a lower bound on the amount by which f must be perturbed in the supremum norm in order to remove all the features (branches and holes) from the MultiNerve Mapper. Conversely,

$$\min_{p \in Dg(\bar{M}_f(X,\mathcal{I}))} d_\infty(p, Q_p^{\mathcal{I}})$$

is a lower bound on the maximum amount of perturbation allowed for f if one wants to preserve all the features in the MultiNerve Mapper no matter what. Note that this does not prevent other features from appearing. The quantity that controls those is related to the points of $Dg(\tilde{f})$ (including diagonal points) that lie in the staircases. More precisely, the quantity

$$\min_{p \in Dg(\tilde{f}) \cup \Delta} d_\infty(p, \partial Q_p^{\mathcal{I}} \setminus \Delta)$$

is a lower bound on the maximum amount by which f can be perturbed if one wants to preserve the structure (set of features) of the MultiNerve Mapper no matter what. Note that this lower bound is in fact zero since $\partial Q_O^{\mathcal{I}} \setminus \Delta$ and $\partial Q_R^{\mathcal{I}}$ come arbitrarily close to the diagonal Δ. This means that, as small as the perturbation of f may be, it can always make new branches appear in the MultiNerve Mapper. However, it will not impact the set of holes if its amplitude is less than

$$\min_{p \in Ext(\tilde{f}) \cup \Delta} d_\infty(p, \partial Q_{E^-}^{\mathcal{I}}).$$

From this discussion, we obtain information about the selection of overlapping between intervals. Having small overlaps between the intervals of the gomic helps capture more features (branches and holes) of the Reeb graph in the (MultiNerve) Mapper; conversely, having large overlaps helps prevent new holes from appearing in the (MultiNerve) Mapper under small perturbations of the function.

Stability w.r.t Perturbations of the Cover

Now, we fix (X, f) and study the case where the gomics are varied. Hence, for each choice of gomic $\mathcal{I}$, we get the following equations:

$$Dg(\bar{M}_f(X, \mathcal{I})) = (Ord_0(f) \setminus Q_O^{\mathcal{I}}) \cup ((Ext_0^+(f) \cup Ext_1^-(f)) \setminus Q_{E^-}^{\mathcal{I}}) \cup (Rel_1(f) \setminus Q_R^{\mathcal{I}})$$
$$Dg(M_f(X, \mathcal{I})) = (Ord_0(f) \setminus Q_O^{\mathcal{I}}) \cup ((Ext_0^+(f) \cup Ext_1^-(f)) \setminus Q_E^{\mathcal{I}}) \cup (Rel_1(f) \setminus Q_R^{\mathcal{I}})$$

tell which points of the diagram $Dg(f)$ end up in the diagram of the (MultiNerve) Mapper and thus participate in its structure. We aim for a quantification of the extent to which this structure may change as the gomic is perturbed.

A distance between gomics. Given a persistence diagram Dg and two gomics $\mathcal{I}, \mathcal{J}$, we consider the quantity:

$$d_{Dg}(\mathcal{I}, \mathcal{J}) = \max_{* \in \{O, E^-, R\}} \{ \sup_{p \in Dg^* \cap (Q_*^{\mathcal{I}} \Delta Q_*^{\mathcal{J}})} max\{d_\infty(p, Q_*^{\mathcal{I}}), d_\infty(p, Q_*^{\mathcal{J}})\}\}, \tag{3.2}$$

where Δ denotes the symmetric difference, where Dg^* stands for the subdiagram of Dg of the right type (Ord, Ext or Rel), and where we adopt the convention that $sup_{p \in \emptyset} \dots$ is zero instead of infinite. Deriving an upper bound on $d_{Dg}(\mathcal{I}, \mathcal{J})$ in terms of the Hausdorff distances between the staircases is straightforward, since

the supremum in (4) is taken over points that lie in the symmetric difference between the staircases:

$$d_{Dg}(\mathcal{I},\mathcal{J}) \leq \max_{*\in\{O,E^-,R\}} d_H(\mathcal{Q}_*^{\mathcal{I}},\mathcal{Q}_*^{\mathcal{J}}),$$

where d_H stands for the Hausdorff distance in the l^∞-norm. The connection to the MultiNerve Mapper appears when we take Dg to be the persistence diagram of the induced map $\tilde{f}$ defined on the Reeb graph $R_f(X)$. Indeed, we have

$$\begin{aligned} Ord(\tilde{f}) \cap (\mathcal{Q}_O^{\mathcal{I}} \Delta . \mathcal{Q}_O^{\mathcal{J}}) &= (Ord(\tilde{f}) \cap \mathcal{Q}_O^{\mathcal{I}}) \Delta (Ord(\tilde{f}) \cap \mathcal{Q}_O^{\mathcal{J}}) \\ &= Ord(\bar{M}_f(X,\mathcal{I})) \Delta Ord(\bar{M}_f(X,\mathcal{J})) \end{aligned}$$

where the second equality follows from the definition of the signature of the MultiNerve Mapper previously given. Similar equalities can be derived with Ext and Rel. Having $d_{Dg(\tilde{f})}(\mathcal{I},\mathcal{J}) = 0$ means that there are no diagram points in the symmetric difference, so the two gomics are equivalent from the viewpoint of the structure of the MultiNerve Mapper. Differently, having $d_{Dg(\tilde{f})}(\mathcal{I},\mathcal{J}) > 0$ means that the structures of the two MultiNerve Mappers differ, and the value of $d_{Dg(\tilde{f})}(\mathcal{I},\mathcal{J})$ quantifies by how much the covers should be perturbed to make the two multigraphs isomorphic. Furthermore, we have the following upper bound on this quantity:

Theorem 3.26. *Given a Morse-type function $f : X \to \mathbb{R}$, for any gomics $\mathcal{I}$, $\mathcal{J}$,*

$$d_{Dg(\tilde{f})}(\mathcal{I},\mathcal{J}) \leq \max_{*\in\{O,E^-,R\}} d_H(\mathcal{Q}_*^{\mathcal{I}},\mathcal{Q}_*^{\mathcal{J}})$$

From this section, we have seen some interesting facts that can be applied to 1-dimensional Mapper. The most outstanding are the following:

(a) As small as a perturbation over a function f may be, we can not assure that we will have the same branches in the MultiNerve Mapper. Indeed, for any amount of perturbation over f, there is the possibility of new branches appearing.

(b) Taking small overlaps between intervals of the gomic helps capture more features of the Reeb graph in the (MultiNerve) Mapper. On the other hand, having large overlaps helps prevent new holes from appearing in the (MultiNerve) Mapper under small perturbations of the function.

(c) For two different gomics $\mathcal{I}$, $\mathcal{J}$, we have obtained an upper bound to quantify by how much the covers should be perturbed to make the two multigraphs isomorphic

Chapter 4

Results

4.1 Introduction

The principal aim of this study is to ascertain the conclusion from *Comparison between endocardial and epicardial cardiac resynchronization in an experimental model of non-ischaemic cardiomyopathy* paper using Topological Data Analysis, *TDA*, tools. In particular, we use the Mapper algorithm to analyze the data set given by the Hospital Sant Pau de Barcelona. Moreover, we are also open to new information that traditional statistics methods could not reflect. Hence, our goal is to contrast the region-dependent response to LV pacing and try to discover other features of the cardiac resynchronization therapy.

We want to remark that this study has been carried out with only a number of six pigs. This fact has its positive and negative aspects. Using statistical methods over a small number of samples can not be really useful, since such a small sample could not reflect the reality. Then, the results obtained applying TDA, in this case the Mapper algorithm, can be much more determinant. On the other hand, with a little amount of studied individuals, there is the possibility of analysing outliers and then the results would also not reflect the reality.

4.2 Dataset

The *Hospital Sant Pau* team analysed the differential effect of endocardial and epicardial pacing on the following variables at each pacing configuration: *LV* peak pressure (*LVP*), *LV dP/dtmax*, *LV dP/dtmin*, mean *ABF*, as well as *QRS* complex width and *QT* interval. The values obtained for each biventricular pacing configuration were compared with those obtained during previous dyssynchronous *RV DDD* pacing at the same *AV* delay.

In other words, they obtained data extracted from six female domestic swine responses, with bipolar pacing electrodes each, against several heart testings. Then, to check the efficiency of the bipolar pacing electrodes, they compared the previous values with those obtained without the bipolar pacing electrodes (*2-basal VD 75 pacing*). These differences were expressed as a percentage of change using the formula:

$$100 \times [(\text{value of variable } X/\text{dyssynchronous value of variable } X) - 1].$$

By applying the previous formula we obtain the variables: ΔLVP, $\Delta LVdP/dtmax$, $\Delta LVdP/dtmin$, ΔABF, ΔQRS and ΔQT. Some of these formulas are the ones we use in our study to check the existence of an improvement using the earlier mentioned method. In particular, we use the variables ΔLVP, $\Delta LVdP/dtmax$, $\Delta LVdP/dtmin$, ΔABF, and notate them as *DPDT+*, *DPDT-*, *LV*, and *FA*, respectively.

Now, we explain the variables we use in the study in detail to facilitate the comprehension throughout the paper for the readers. The four variables compare the results obtained using bipolar pacing electrodes and without them. Hence, considering that the variables represent a comparison, we explain every feature that has been compared.

(a) *DPDT+*. This variable reflects the maximal rate of rising left ventricular pressure (LVP).

(b) *DPDT-*. Samely to *DPDT+*, this one indicates the minimal rate of rising left ventricular pressure (LVP).

(c) *LV*. It represents the left ventricular pressure.

(d) *FA*. This variable stands for arterial blood flow.

Finally, we want to specify that we work with 576 points. There are three regions (*base*, *media*, and *apical*), and each of them has subregions. In particular, base and mid have the same three subzones; *posterior*, *anterior* and *lateral*. On the other hand, the *apical* region has two other subzones (*apical1* and *apical2*). Moreover, endocardial and epicardial pacing is differentiated for these eight regions. Then, we have an amount of 16 labels for every swine.

Additionally, the medical team used six different machine configurations to analyse all the mentioned labels above. Hence, we have a total of 96 points for every swine. So, all the points of all pigs sum 576 points, as we stated at the beginning.

4.2.1 Data visualization

Now, we try to understand and get information about the data by visualizing the point cloud. Hence, we have developed a Python program[1] to plot the mentioned variables in $\mathbb{R}^3$. However, we also study the planes between variables since, in general, it provides a more clear visualization of the distinguished little cluster and variables regression.

In this study, three main labels are used during the process. For every plot, we use colours to differentiate between three labels that can give us some notable information. These labels compare the endocardial and epicardial pacing, different heart regions, and the six individuals. The heart has been divided into basal, mid, and apical zones, so the heart regions' label is made out of them.

Thus, we divide this section into three subsections, one for each label, and we put the most outstanding plots[2]. All three sections ahead follow the same representation pattern. There is just one figure in each section. In every figure, we find 12 planes where all the values are reflected. The axes of every plane are two out of the four we use to analyse the data, so we can also understand the relation between them. This will also be reflected in the third section, where we implement a PCA analysis. However, the represented colours may vary from section to section depending on the reflected label.

Endocardial vs Epicardial

The following plots show the values differentiated by the endocardial and epicardial comparison. We map the epicardial points to blue and the endocardial ones to red colour for this label, and the obtained graphs are represented on the next page:

[1] You can check the complete code in **[POSAR EL NUMERO DEL ANNEX]**

[2] For all the created plots check the annexe.

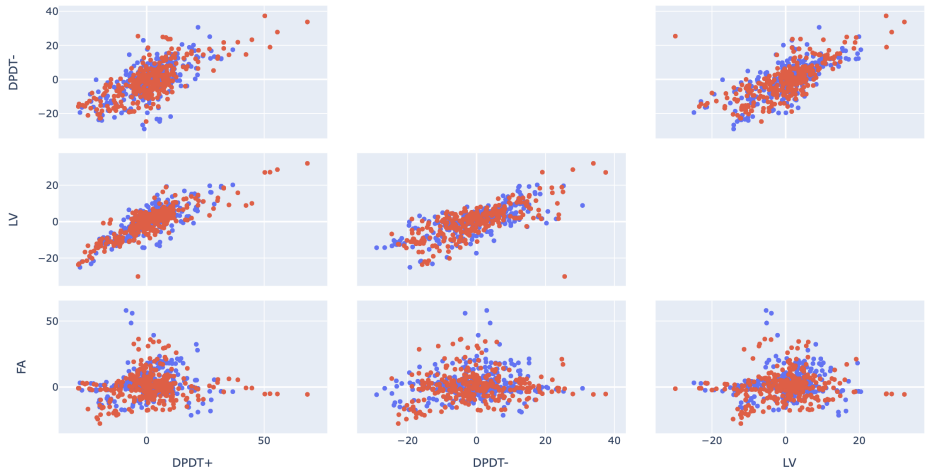

Figure 4.1: *Plot of the data in* $\mathbb{R}^2$ *differentiated by epicardial (blue) and endocardial (red) pacing.*

Regions (Basal, Mid, Apical)

Here, we reflect on the three different heart zones, i.e. the Basal, Mid, and Apical regions. The colours for the next figure are mapped in the following way: blue to basal, red to mid, and green to apical.

Figure 4.2: *Plot of the data in* $\mathbb{R}^2$ *differentiated by the heart's regions; basal (blue), mid (red) and apical (green).*

Swines

Basically, we assign a color to each swine, and the most important plots are:

Figure 4.3: *Plot of the data in* $\mathbb{R}^2$ *differentiated by swines.*

4.2.2 Conclusions

As we can imagine, there is a noticeable linear correlation between the variables *DPDT+*, *DPDT-* and *LV* for all three labels. However, there is nothing we can distinguish from the variable *FA* in that sense.

Note that every plot follows a similar pattern, i.e. there is a big centred cluster where the vast majority of the points lie, and then some other small clusters made out of a few points.

In subsection the regions plots, the small clusters out of the centre only have basal points or a mix of mid and apical. Hence, since one of our goals is to distinguish between basal to mid and apical regions, this fact can be determinant in the implementation of Mapper.

Furthermore, in the epicardial and endocardial figures, we can also find small clusters of just endocardial or epicardial pacing points. Nevertheless, we can not compare the size and frequency of these clusters to those in the regions planes.

In the other section, we can observe a differentiation of the *FIS13* and *FIS6* to the other pigs, although we can also find some separated clusters of *FIS18* and *FIS14*.

Notice that, for the variables *DPDT+*, *DPDT-* and *LV*, all the clusters mentioned

above are aligned with the central one. Then, they don't break the regression, only have considerable higher values, so we are not dealing with outliers.

In conclusion, we have visualised some meaningful results even before filtering the data. Thus, we can be optimistic about obtaining determinant information after applying Mapper with a proper filter function and cluster algorithm, mainly for the heart regions.

4.3 *PCA* analysis

As stated before, we analyse our dataset through PCA in this subsection. Firstly, we show the different plots of our data in the planes where the principal components are the axes. There are three different figures for every label we have, as we did in previous section. Nevertheless, we have decided to show the plots for the data filtered by heart regions since it is the only one we can mention relevant information about[3].

Furthermore, in this section, there are two other figures extracted from the principal components analysis that provide information about the relation of the original variables and the new set of variables, the principal components.

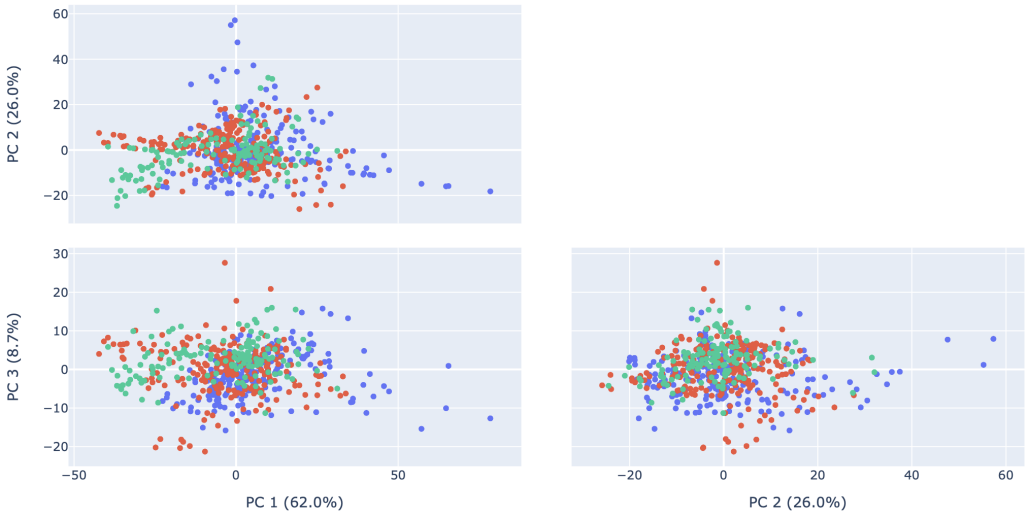

Figure 4.4: *Plot of the first three Principal Components in* $\mathbb{R}^2$ *differentiated by heart's zones; basal(blue), mid (red), and apical (green).*

[3]Check the plots for an epicardial and endocardial comparison, and for the different swines in the Annex.

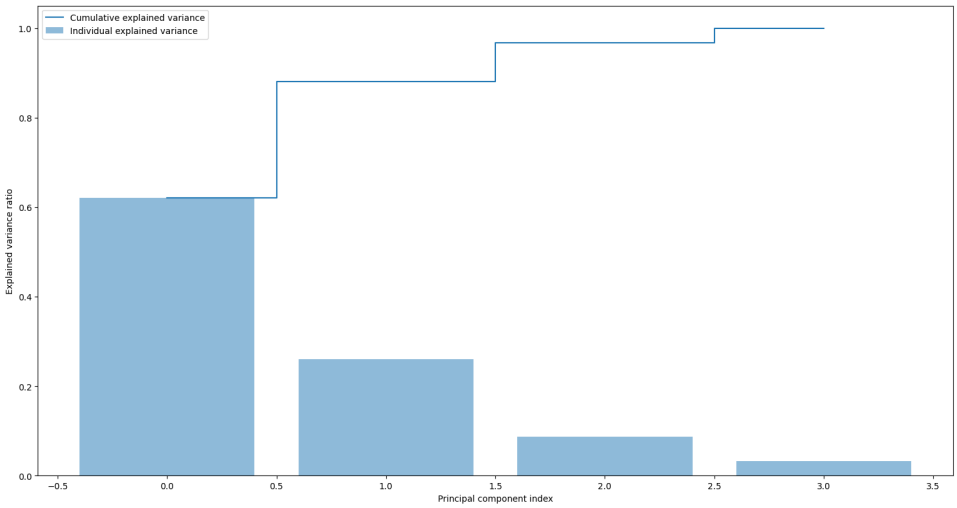

Figure 4.5: *Explained Variance histogram. Each column shows the amount of explained variance of the first kth principal component, and the line represents the cumulative explained variance of the first k principal components.*

By studying the three obtained pictures, we can state that the first and second principal components have an 88% weight over the data information. Hence, since it is representative enough, we use *PCA* with two components as our filter function for Mapper.

From now on, we focus on the planes with the first and second principal components as axes. Thus, in the plot labelled by the heart zones, we can distinguish two clusters, made out of basal points, from the centre, where most of the points lie. Also, there is an apical cluster, even though it is closer to the centre, and some mid-region points are around.

On the other hand, apparently, there are no noticeable distinctions in the epicardial and endocardial plots. Thus, it seems pretty challenging to think about getting new information from this label, although we are still open to finding new information using Mapper. Nevertheless, this ascertains the hypothesis obtained by *Hospital Sant Pau de Barcelona* related to this topic. Specifically, endocardial LV pacing induces similar haemodynamic changes to pacing from the epicardium.

Finally, from the swine differentiated plot, we can state some slight differences between the swines *FIS13* and *FIS6* from the rest. Also, some *FIS15* points are out of the centre, although quite close. Hence, we can not say that the *FIS15* case is not as straightforward as the two mentioned before.

Furhtermore, we can mention that the variables *DPDT+*, *DPDT-* and *LV* have similar directions, but this makes sense because of the earlier linear regressions. By analogous reasoning, the variable FA has a different direction than the rest. Then, as we mentioned before, the first and second principal have a significant weight over the dataset information, particularly 88%.

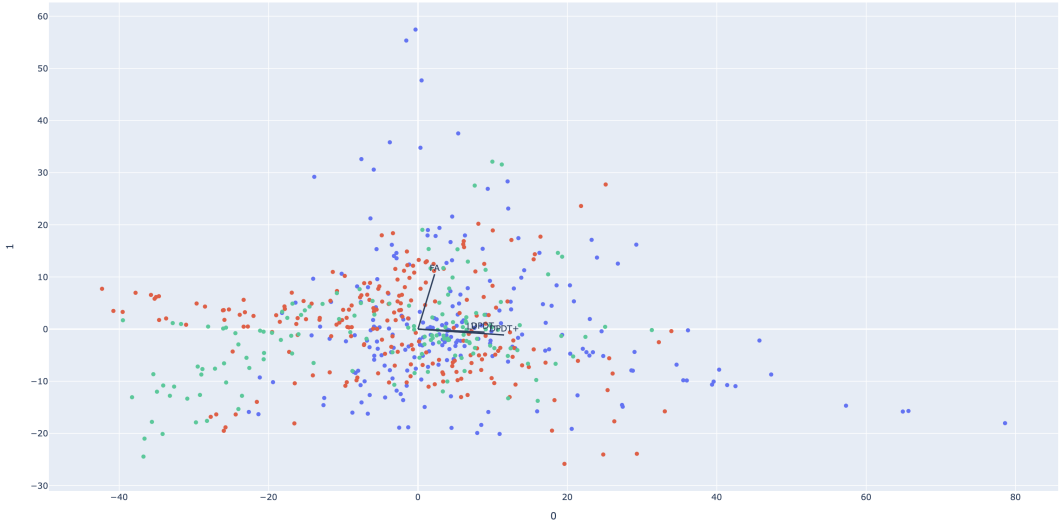

Figure 4.6: *Plot of the filtered values by PCA differentiated by heart's zones, and variables directions over the principal components.*

4.4 Applying Mapper algorithm

Now that we have already decided on the most suitable filter function for our case, we apply the Mapper in this section. However, we still have to determine the clustering algorithm, the intervals and their overlapping. We represent the results for the best clusterer[4] for our case. However, we will keep varying the other two parameters to prove some consistency in our results.

The selected clustering algorithm is the K-Means algorithm from the *sklearn* library. After trying the most common and useful clusterers, and comparing the results, we saw that K-Means was the most suitable option, even though there were repeated patterns in some of them.

It might be fair to remark that K-Means is not entirely stable. The graphs obtained are not exactly the same, but their differences are almost insignificant. Specifically, the values obtained for the variables nodes, total samples, and unique samples do not change from one graph to the other. However, the number of edges in the graph tends to vary a little. Then, even though this change may be pretty negligible, it can slightly affect the visualisation of the final result.

However, even though the clustering algorithm is not entirely stable, it is also fair to remark that the differentiation of the basal region was visible in every graph. Then, it is evident that we have chosen the figures with the most considerable differentiations after running the Python program a few times for each set of parameters[5].

We want to remark that we have identified the basal points with yellow, mid with green, and apical with purple. Then, each node has associated the color of the dominant region. However, if the representation of two colors is the same then it defines a new color in between of the two regions.

Now, we see the plots given by the Mapper algorithm with PCA and K-Means as their lens and clustering algorithm parameters. On the other hand, as we said earlier, we show several figures where we have changed the number of intervals and their overlapping to prove some consistency in the final results.

[4]For all the filter functions analysis check the annexe.

[5]You can check other graphs with the same parameters in the annexe of this paper.

4.4.1 Mapper Results

Figure 4.7: *Mapper graph obtained with seven intervals and* 35% *overlapping percentage between intervals.*

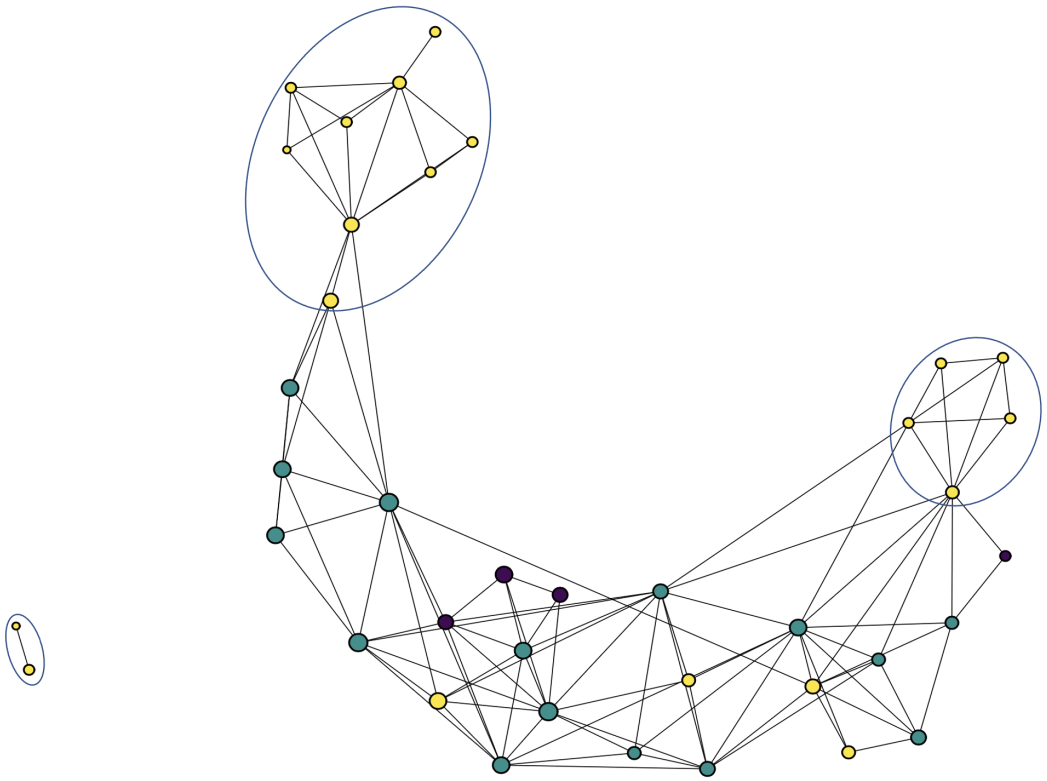

Figure 4.8: *Mapper graph obtained with four intervals and* 42.5% *overlapping percentage between intervals.*

4.5 Quantification in Graphs

In this section, we give arguments through basic graph theory to justify the results obtained from Mapper quantitatively.

Firstly, we want to remark that the information one can extract from a Mapper graph is purely from the connectivity between vertices. Hence, the distance between vertices or the distribution of the graph is relevant. However, getting visually attractive shapes helps to understand the information from the graph more easily.

Let $G = (V, E)$ be a graph, where V is the set of vertices and E is the set of edges. The elements of V, the vertices, are denoted by v_i, and the edges, elements of E, by $e_{i,j} = (v_i, v_j)$ such that $i \neq j$. We want to distinguish between the vertices depending on their colours; hence, we need to define the following vertices subsets:

$$V_Y = \{v_i \in V : v_i \text{ is yellow}\}, V_G = \{v_i \in V : v_i \text{ is green}\},$$
$$\text{and } V_P = \{v_i \in V : v_i \text{ is purple}\}.$$

Furthermore, we also need to define some edges subsets:

$$E_Y = \{e_{i,j} = (v_i, v_j) : v_i \in V_Y, v_j \notin V_y\}, E_G = \{e_{i,j} = (v_i, v_j) : v_i \in V_G, v_j \notin V_G\}$$
$$\text{and } E_P = \{e_{i,j} = (v_i, v_j) : v_i \in V_P, v_j \notin V_P\}.$$

Notice that these sets are defined by the edges that connect different coloured vertices. Then, the first characteristic of our graph that we have studied is given by:

$$\bar{X}_{DC,i} = \frac{|E_i|}{|V_i|}, \text{ for } i \in \{Y, G, P\}.$$

Intuitionally, for a given color i, we calculate the mean of the amount of edges $e_{(i,j)}$ such that $i \neq j$. Now, let us define the following subsets:

$$V_{YY} = \{v_i \in V_Y : \nexists\, e_{i,j} = (v_i, v_j) \text{ s.t. } v_j \in (V \setminus V_Y)\},$$
$$V_{GG} = \{v_i \in V_G : \nexists\, e_{i,j} = (v_i, v_j) \text{ s.t. } v_j \in (V \setminus V_G)\},$$
$$\text{and } V_{PP} = \{v_i \in V_P : \nexists\, e_{i,j} = (v_i, v_j) \text{ s.t. } v_j \in (V \setminus V_P)\}.$$

We have defined the set of vertices that either have edges connecting them to only vertices of the same colour or are not connected. Thus, we have studied the following characteristic:

$$f_{MC,i} = \frac{|V_{ii}|}{|V_i|}, \text{ for } i \in \{Y, G, P\}.$$

We can say that $f_{MC,i}$ defines, for each color, the relative frequency of vertices in V_{ii} over the set V_i, for $i \in \{Y, G, P\}$. In other words, for each colour, $f_{MC,i}$ defines the relative frequency of vertices that are not connected to other coloured vertices over the total amount of vertices of the respective colour.

Now, calculating $\bar{X}_{DC,i}$ and $f_{MC,i}$ over the two illustrated graphs, we obtained the following results:

	Intervals = 4, Overlapping = 42.5%		Intervals = 7, Overlapping = 35%	
	$\bar{X}_{DC}$	f_{MC} (%)	$\bar{X}_{DC}$	f_{MC} (%)
YELLOW	1.5	60%	0.9	66%
GREEN	2.73	13.33%	2.07	20.69%
PURPLE	3	0%	2.33	11.11%

Figure 4.9: *Table with the $\bar{X}_{DC}$ and f_{MC} values for each obtained Mapper graph.*

Observing the table above, we can affirm the following statements:

(a) For both graphs, note $\bar{X}_{DC,Y}$ is the half, or almost half, of $\bar{X}_{DC,G}$ and $\bar{X}_{DC,P}$. Then, the connectivity with yellow vertex with different coloured vertices is lower than the rest.

(b) Moreover, the values for $f_{MC,Y}$ are much higher compared to ones for $f_{MC,G}$ and $f_{MC,P}$. The previous fact implies that the yellow vertices' tendency to connect only to vertices of their same colour is greater than for the green and purple vertices.

4.6 Mapper conclusions

From the above graphs, and their quantification results, we can appreciate a difference between the basal region and the two others (mid and apical). In all shown figures, each with different interval numbers and overlapping percentages, there are some clusters where most of their samples are from the heart basal region. Notice that, in general, this does not happen with any zone in such a clear way.

Hence, we think one of the leading hypotheses we established at the beginning of this study has been accomplished. The hypothesis we are talking about is the following one:

The response of epicardial and endocardial LV pacing was regional dependent and the best response was obtained at the basal regions.

Furthermore, we get more great graphs when filtering the dataset via endocardial values than for epicardial pacing. Even though we can get differences for both of them, the frequency of obtaining highlighting graphs for endocardial is higher. Thus, we can state that it can be a slight improvement in the therapy through endocardial pacing.[6]

[6]Check some Mapper graphs sorted by endocardial and epicardial values in the annex of this paper.

4.7 Contrast by Statistical Methods

We have applied statistical methods to our dataset to compare and contrast the results obtained by the Mapper algorithm. We have computed the mean, standard deviation, standard error, and maximum and minimum value of our data for the entire data. Then, we have studied the mean, standard deviation and standard error differentiating between epicardial and endocardial pacing. The results are the following:

HEART	REGIONS			
	DPDT+	DPDT-	LV	FA
		MEAN ± SEM		
BASAL	7.076567 ± 2.763061	1.840348 ± 2.517819	3.208417 ± 1.633373	2.926700 ± 2.956022
MID	-0.465862 ± 2.686426	-2.110055 ± 2.182487	-0.631245 ± 2.006976	1.339884 ± 2.134980
APICAL	-3.245893 ± 2.579310	-0.889392 ± 2.491564	-1.961357 ± 1.832959	-0.613629 ± 2.333782
		STD		
BASAL	11.722677	10.682203	6.929815	12.541340
MID	11.397541	9.259510	8.514878	9.057954
APICAL	10.943085	10.570810	7.776586	9.901399
		MIN		
BASAL	-10.661261	-19.537381	-11.259651	-17.659084
MID	-29.072220	-28.938618	-29.929461	-21.374188
APICAL	-27.551760	-23.920343	-21.817713	-27.643564
		MAX		
BASAL	68.461892	37.571636	32.269714	58.105418
MID	30.373618	25.653401	16.408424	32.450977
APICAL	28.805269	23.890820	14.757875	34.621425

Figure 4.10: *Table of statistical methods values sorted by heart regions.*

EPI				
	DPDT+	DPDT-	LV	FA
		MEAN ± SEM		
BASAL	5.533626 ± 2.083743	1.634607 ± 2.308975	3.294722 ± 1.379208	4.088982 ± 3.062976
MID	1.281838 ± 2.439489	-0.857306 ± 2.272587	0.402796 ± 1.897390	2.327843 ± 2.197575
APICAL	-3.229637 ± 2.655434	-0.953488 ± 2.388040	-1.876403 ± 1.885841	-0.180911 ± 1.660412
		STD		
BASAL	8.840574	9.796153	5.851482	12.995108
MID	10.349877	9.641771	8.049944	9.323521
APICAL	11.266054	10.131594	8.000947	7.044533

Figure 4.11: *Table of statistical methods values sorted by heart regions via epicardial pacing.*

Analysing the content given in the three tables, we can appreciate some relevant information about the dataset. Observe a noticeable difference between the basal mean and the others. However, notice that there are high standard deviations for all regions and variables. This is consequence of the huge distances between the maximum and minimum values. Then, we mostly base our comparison using the mean and standard error (SEM) together.

ENDO				
	DPDT+	DPDT-	LV	FA
		MEAN ± SEM		
BASAL	8.619508 ± 3.275880	2.046089 ± 2.720655	3.122112 ± 1.859489	1.764419 ± 2.832750
MID	-2.213562 ± 2.864883	-3.362804 ± 2.056784	-1.665286 ± 2.091302	0.351925 ± 2.054283
APICAL	-3.262150 ± 2.519528	-0.825297 ± 2.607685	-2.046312 ± 1.791532	-1.046347 ± 2.862119
		STD		
BASAL	13.898383	11.542760	7.889145	12.018340
MID	12.154670	8.726196	8.872645	8.715585
APICAL	10.689451	11.063470	7.600826	12.142941

Figure 4.12: *Table of statistical methods alues sorted by heart regions via endocardial pacing.*

First, we examine the values from the general table, the one without differentiation between endocardial and epicardial pacing. For the variables DPDT+ and LV, we get that the intersection of the basal interval and the mid interval in this table is empty. Moreover, by comparing the basal and apical, we also obtain that the variables DPDT+ and LV are the only ones with an empty intersection between intervals. Contrarily, there is no empty intersection for any variable comparing mid and apical.

Now, we examine the table with only endocardial pacing values. Comparing the basal and mid intervals, we have empty intersections for the variables DPDT+, DPDT- and LV. Then, we can appreciate empty intersections for DPDT+ and LV for basal and apical. Again, we can not say anything about the comparison between apical and mid regions.

Finally, we analyse the epicardial values table. We can not tell anything about any differentiation (empty intersections) between basal and mid regions for epicardial pacing. However, we get an empty intersection for the variables DPDT+ and LV by comparing basal and apical. Same as the other two tables, there is nothing relevant to highlight comparing mid and apical regions.

Thus, we can conclude that for some variables, there is a noticeable differentiation between basal and mid, or basal and apical. However, there is nothing we can say about comparing the mid and apical regions. These results contrast the conclusions previously obtained by the Mapper algorithm since we can observe some differences from the basal region.

More profoundly, the endocardial pacing has more empty intersections than the epicardial pacing. Then, this reflects the fact that, for endocardial values, the Mapper gets graphs where the basal is differentiated more frequently.

Chapter 5

Conclusions

This study was carried out to complement and enhance the results abstracted by the Department of Cardiology from the *Hospital de la Santa Creu i Sant Pau*. We expected to find apparent differences in the different heart regions by studying the improvement resulting from the heart responses after being submitted to bipolar pacing electrodes. Furthermore, we were also confident about discovering significant results related to other features.

We began the paper by building a theoretical framework that would help the reader to get a fully detailed understanding of the methods and tools used later in our study. Throughout this construction, we have stated results both from topology and statistics. Firstly, an explanation of the Nerve Theorem, the core of the Mapper algorithm, was given with all the topological concepts needed to comprehend it. On the other hand, we also dedicated a section to *PCA* and its differences from *SVD*. Furthermore, we explained Mapper's methodology and offered a proof of the relation between the Nerve Theorem and our Mapper implementation. Finally, we stated some results about the stability of 1-dimensional Mapper via its connection to Reeb graphs.

The practical part started by explaining exhaustively the dataset provided by the Cardiology Department. We described the methods they followed to obtain the data, the displayed variables and their meaning. Moreover, we detailed the three labels we were willing to study and compare to abstract relevant results.
Afterwards, we reflected the data into planes taking two of the four variables we had as their axes. This visualisation allowed us to study the correlation between variables. Hence, a clear correlation was abstracted by comparing the *DPDT+*, *DPDT-* and *LV* variables, although nothing relevant was obtained for *FA*.
Later on, a principal components analysis was implemented into our dataset. From the analysis, we could obtain significant information, but mainly it was

helpful to contrast that the two principal components were the most optimal filter function in our case. We got to this decision after checking that the two principal components had an 88% weight over the data information and visualising the plots of the data filtered by them.
Finally, we provided the most outstanding results of the Mapper application in the initial data. In order to supplement the results obtained through the algorithm, we carried out a study using statistical methods and a quantification of the graphs using some basic graph theory concepts. In particular, we differentiated the basal region from the mid and apical. Hence, we can assert that there is a better improvement of the bipolar pacing electrodes if they are placed in the basal region of the heart. However, we could not get any relevant differentiation from the other labels.

Mainly, we showed the best graphs that reflected the hypothesis and goals we wanted to contrast in the introduction. However, more results can be checked in the Annex of this paper.

Even though we could not provide new information, we still offered more reasons to justify differences in the heart' zones. Since both studies have concluded with the same results, it indeed can be expected that there is a better response in the heart's basal. Hence, we are still motivated to keep studying this data by applying other *TDA* methods, such as persistent homology, to complement these studies.

We hope that all this work has a meaningful impact on the treatment of arrhythmias and is useful to future readers to clarify some aspects of the Mapper Algorithm. Indeed, we encourage these readers to go further in this research and get valuable results for the theory behind Mapper.

Bibliography

[1] Li, L., Cheng, W. Y., Glicksberg, B. S., Gottesman, O., Tamler, R., Chen, R., Bottinger, E. P., & Dudley, J. T., (2015), *Identification of type 2 diabetes subgroups through topological analysis of patient similarity*, Science translational medicine, 7(311), 311ra174. https://doi.org/10.1126/scitranslmed.aaa9364

[2] Chazal, F. and Michel, B., *An Introduction to Topological Data Analysis: Fundamental and Practical Aspects for Data Scientists*, arXiv preprint arXiv:1710.04019, (2017).

[3] Hatcher, Allen, *Algebraic topology*, Cambridge: Cambridge University Press, (2002).

[4] Cavanna, Nicholas J., and Donald R. Sheehy *The generalized persistent nerve theorem*, arXiv preprint arXiv:1807.07920, (2018).

[5] Trefethen, Lloyd Nicholas and Bau, David. Numerical Linear Algebra. Philadelphia: SIAM, (1997).

[6] Jolliffe, I.T., *Principal Component Analysis (2nd ed)*, Springer Verlag, (1986).

[7] Michael, E., *Another Note on Paracompact Spaces*, Proceedings of the American Mathematical Society 8, no. 4 (1957): 822-28, https://doi.org/10.2307/2033306.

[8] Mathieu Carriere, Steve Y. Oudot *Structure and Stability of the 1-Dimensional Mapper*, Foundations of Computational Mathematics, Springer Verlag, (2017), pp.1-64. 10.1007/s10208-017-9370-z , hal- 01633101v2.

[9] Amorós-Figueras G, Jorge E, Raga S, Alonso-Martin C, Rodríguez-Font E, Bazan V, Viñolas X, Cinca J, Guerra JM, *Comparison between endocardial and epicardial cardiac resynchronization in an experimental model of non-ischaemic cardiomyopathy*, Europace, (2018) Jul 1;20(7):1209-1216. doi: 10.1093/europace/eux212. PMID: 29016778.

[10] Kraft, Rami, *Illustrations of Data Analysis Using the Mapper Algorithm and Persistent Homology*, (2016).

[11] Singh, Gurjeet Kaur Chatar, Facundo Mémoli and Gunnar E. Carlsson, *Topological Methods for the Analysis of High Dimensional Data Sets and 3D Object Recognition*, PBG@Eurographics (2007).

[12] Chazal, Frédéric, and Bertrand Michel, *Covers and nerves: union of balls, geometric inference and Mapper*, INRIA, Barcelona (2016).

[13] Dey, Tamal K., Facundo Mémoli, and Yusu Wang, *Multiscale mapper: Topological summarization via codomain covers*, Proceedings of the twenty-seventh annual acm-siam symposium on discrete algorithms, Society for Industrial and Applied Mathematics, (2016).

[14] Frédéric Meunier, Luis Montejano, *Different versions of the nerve theorem and colourful simplices*, Journal of Combinatorial Theory, Series A, Elsevier, (2019). hal-03247121

[15] Shlens, Jonathon, *A tutorial on principal component analysis*, arXiv preprint arXiv:1404.1100 (2014).

[16] Mathieu Carriere, Bertrand Michel, Steve Y. Oudot, *Statistical analysis and parameter selection for Mapper*, Journal of Machine Learning Research, Microtome Publishing, (2018), hal-01633106v2

[17] van Veen et al., *Kepler Mapper: A flexible Python implementation of the Mapper algorithm*, Journal of Open Source Software, 4(42), 1315, (2019), https://doi.org/10.21105/joss.01315

[18] Hendrik Jacob van Veen, Nathaniel Saul, David Eargle, & Sam W. Mangham,. *Kepler Mapper: A flexible Python implementation of the Mapper algorithm (v2.0.1)*, Zenodo, (2021), https://doi.org/10.5281/zenodo.4754451

Annex

A.1 Filter Functions

There are many possibilities when choosing our filter function for Mapper. In particular, we can use many projection functions from maths, statistics, econometrics, or machine learning. Moreover, we can also make combinations between them, so we do not have to stay with just 1-dimensional lenses. A list of all the tested filter functions and some visual examples are given below:

(a) **km.KeplerMapper().fit_transform(X, projection='__')**. Projection parameter is either a string, a *Scikit-learn* class with $fit\_transform$, or a list of dimension indices.

(b) **sklearn.manifold.TSNE(n_components=3, init='pca', perplexity = 75, metric = 'euclidean', n_iter = 5000).fit_transform(X)**. The parameter *metric* is the metric to use when calculating distance between instances in a feature array, some examples are: '*braycurtis*', '*canberra*', '*chebyshev*', '*cityblock*', '*correlation*', '*cosine*' and '*euclidean*'.

(c) **sklearn.manifold.MDS(n_components=2, metric = '__').fit_transform(X)**

(d) **sklearn.manifold.SpectralEmbedding(n_components=2, affinity = '__')**. It forms an affinity matrix given by the specified function and applies spectral decomposition to the corresponding graph laplacian. The function to specify can be one of the following ones: '*nearest_neighbors*', '*rbf*', '*precomputed*', '*precomputed_nearest_neighbors*'.

(e) **sklearn.manifold.LocallyLinearEmbedding(n_components=2).fit_transform(X)**

(f) **sklearn.manifold.Isomap(n_components=2).fit_transform(X)**

(g) **mapper.filters.Gauss_density(X, sigma = 10, metricpar=, callback=None)**

(h) **mapper.filters.eccentricity(X, exponent=1.0, metricpar=, callback=None)**

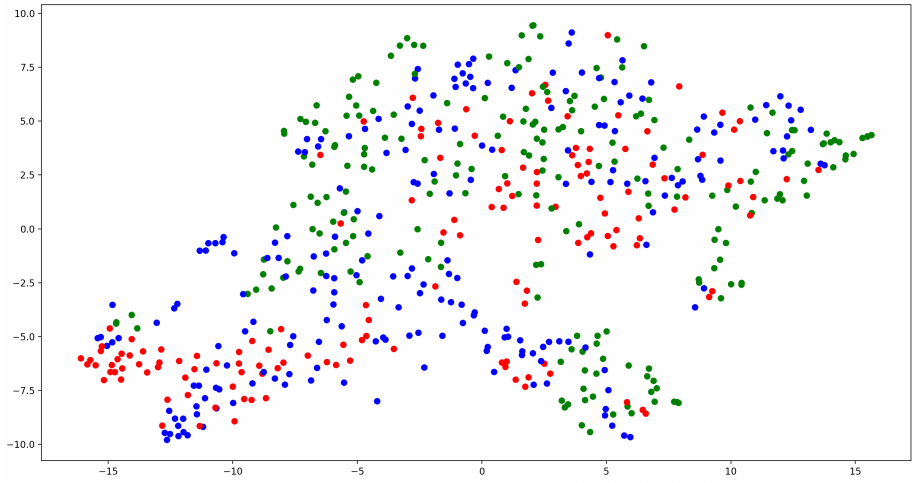

Figure 1: Data filtered by *sklearn.manifold.TSNE*

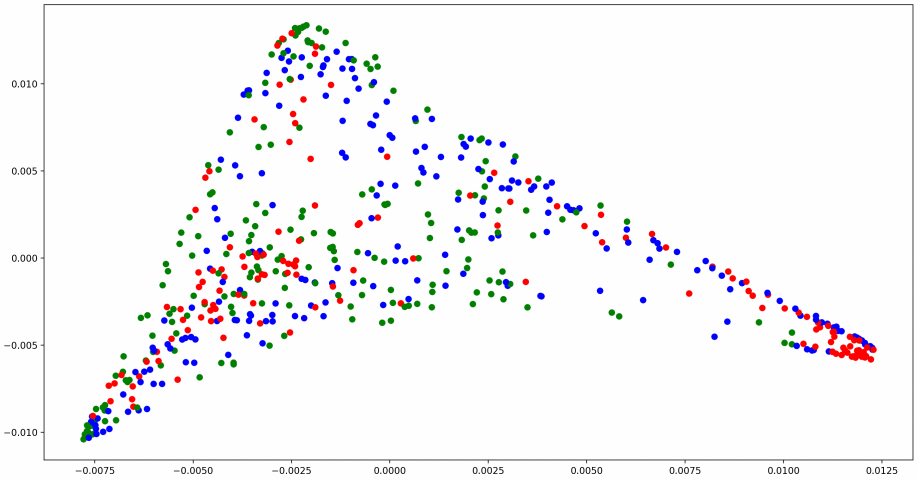

Figure 2: Data filtered by *sklearn.manifold.SpectralEmbedding*

A.2 Clustering Algorithms

Similarly to the filter functions section, we will also list the clustering algorihtms we have tested when applying Mapper to our dataset. The list is as follows:

(a) **sklearn.cluster.DBSCAN(eps=1.4, min samples=3)**.

(b) **sklearn.cluster.AgglomerativeClustering(n clusters=3)**.

(c) **sklearn.cluster.AffinityPropagation(damping = 0.8665)**

(d) **sklearn.cluster.Birch(threshold=0.000001, n clusters=3)**.

(e) **sklearn.cluster.MiniBatchKMeans(n clusters = 3)**

(f) **sklearn.cluster.MeanShift()**

(g) **sklearn.cluster.OPTICS(eps=4.5, min samples=4)**

(h) **sklearn.cluster.SpectralClustering(n clusters=3)**

After comparing the results obtained, with several parameter configurations, by this clustering algorithms with the *K-Means* algorithm we concluded that using the last was the optimal selection in our case.

A.3 Extra PCA plots

Now, we show the plots where we illustrate our dataset in a plane that takes the principal components as its axes. The first figure in this section corresponds to the dataset filtered by epicardial and endocardial pacing. As stated previously in the paper, the epicardial points are mapped to blue and the endocardial to red. Then, in the other figure, we filter the data by the different swines, each with a different colour.

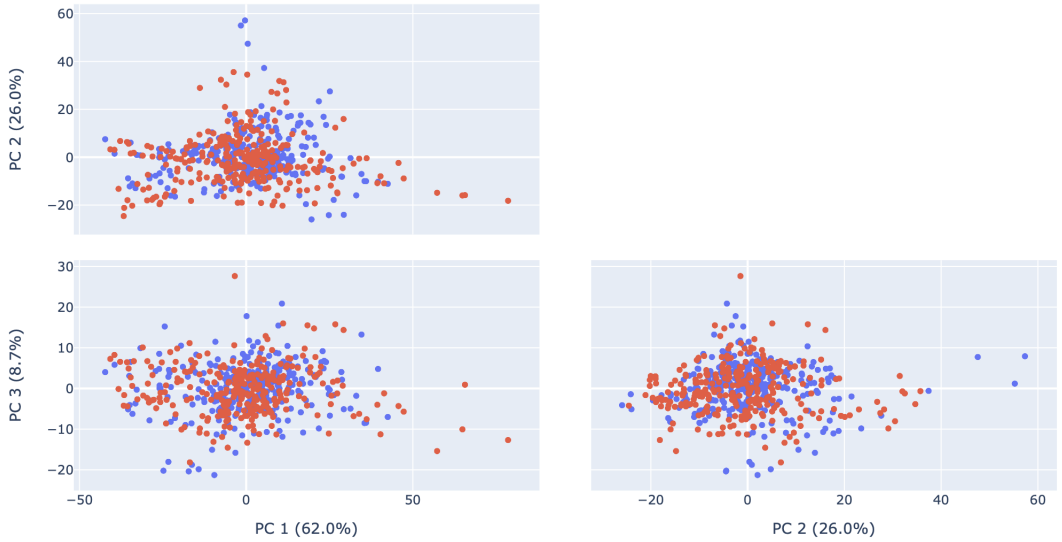

Figure 3: *Plot of the first three Principal Components in* $\mathbb{R}^2$ *differentiated by epicardial (blue) and endocardial (red).*

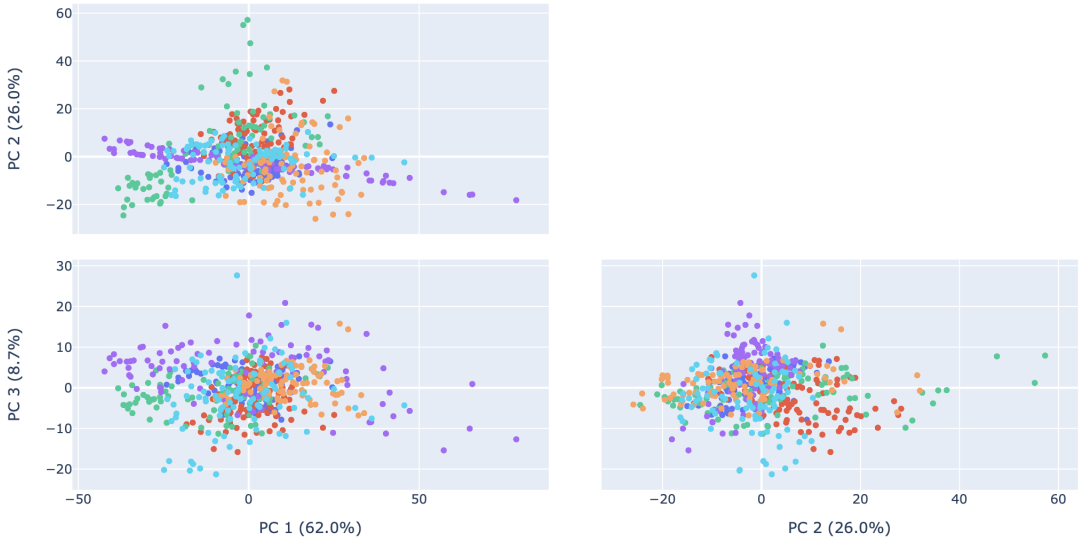

Figure 4: *Plot of the first three Principal Components in* $\mathbb{R}^2$ *differentiated by swines.*

A.4 Extra Mapper plots

In this section, we illustrate more plots where it can be possible to differentiate between basal values from the rest. The first two figures are the obtained results for the whole dataset, but with different values for the number of intervals and overlapping percentages. Moreover, we will see an extra example for endocardial pacing with different parameters. Finally, even though it is not as straightforward as in general or endocardial values, we show the most illustrative graph for epicardial pacing.

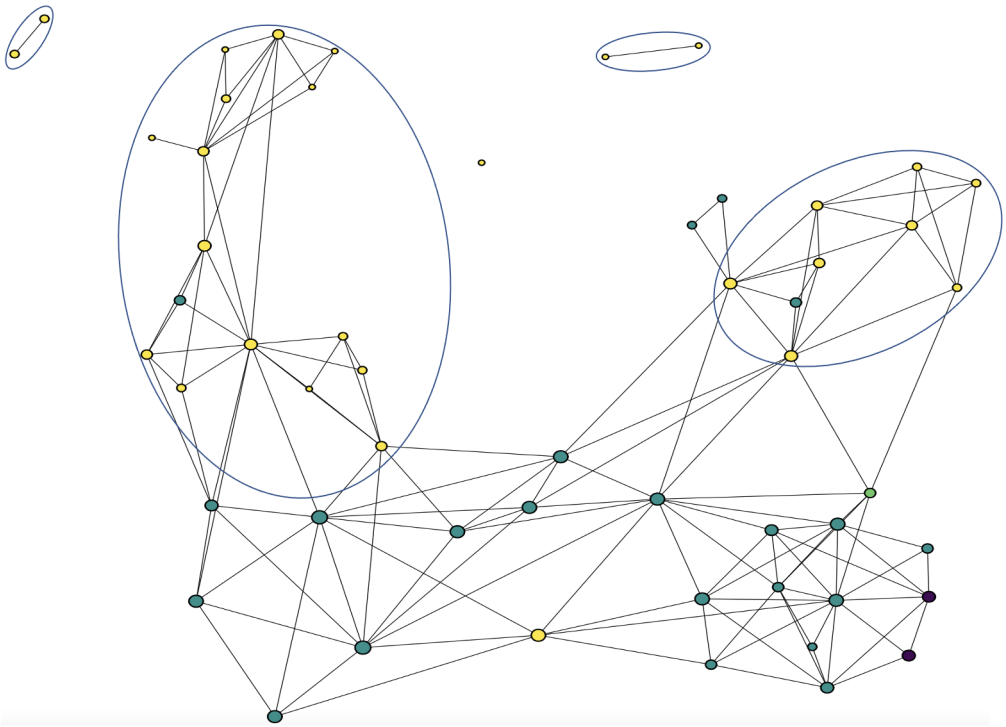

Figure 5: *Mapper graph obtained with six intervals and* 42.5% *overlapping percentage between intervals.*

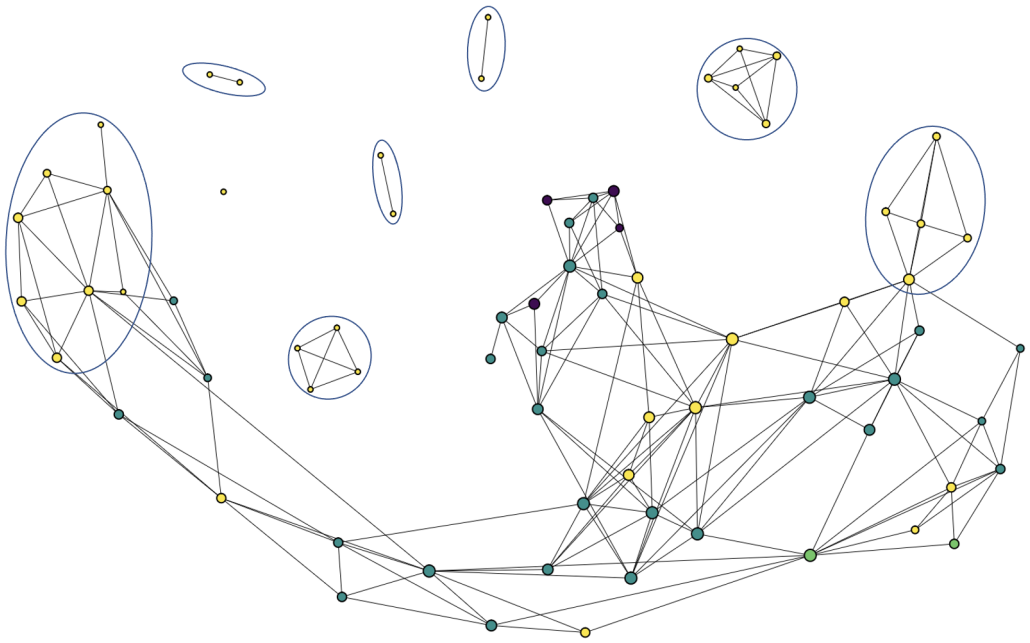

Figure 6: *Mapper graph obtained with six intervals and* 40% *overlapping percentage between intervals.*

The following figures show outstanding Mapper results applied to our dataset but previously filtered by endocardial and epicardial values. There are two plots for endocardial and just one for epicardial since, as mentioned earlier, the frequency is great graphs is higher for the first one.

Figure 7: *Mapper graph obtained with seven intervals and* 40% *overlapping percentage between intervals applied to data filterd by endocardial values.*

Figure 8: *Mapper graph obtained with six intervals and* 45% *overlapping percentage between intervals applied to data filterd by endocardial values.*

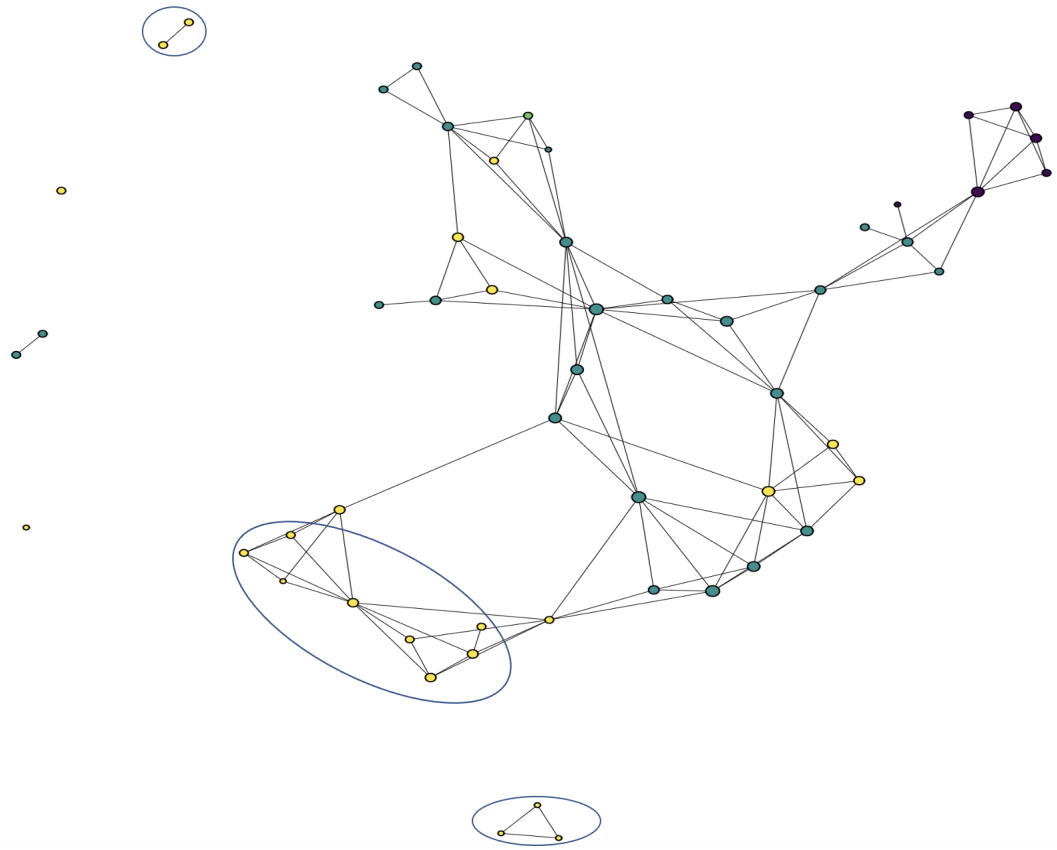

Figure 9: *Mapper graph obtained with five intervals and* 40% *overlapping percentage between intervals applied to data filterd by epicardial values.*