



저작자표시-비영리-변경금지 2.0 대한민국

이용자는 아래의 조건을 따르는 경우에 한하여 자유롭게

- 이 저작물을 복제, 배포, 전송, 전시, 공연 및 방송할 수 있습니다.

다음과 같은 조건을 따라야 합니다:



저작자표시. 귀하는 원저작자를 표시하여야 합니다.



비영리. 귀하는 이 저작물을 영리 목적으로 이용할 수 없습니다.



변경금지. 귀하는 이 저작물을 개작, 변형 또는 가공할 수 없습니다.

- 귀하는, 이 저작물의 재이용이나 배포의 경우, 이 저작물에 적용된 이용허락조건을 명확하게 나타내어야 합니다.
- 저작권자로부터 별도의 허가를 받으면 이러한 조건들은 적용되지 않습니다.

저작권법에 따른 이용자의 권리는 위의 내용에 의하여 영향을 받지 않습니다.

이것은 [이용허락규약\(Legal Code\)](#)을 이해하기 쉽게 요약한 것입니다.

[Disclaimer](#)

Thesis for the Degree of Master of Science

Application of Topological Data Analysis for Biomechanical Dataset

Sangman Jung

Department of Mathematics

Graduate School

Kyung Hee University

Seoul, Korea

February, 2021

Application of Topological Data Analysis for Biomechanical Dataset

Sangman Jung

Department of Mathematics

Graduate School

Kyung Hee University

Seoul, Korea

February, 2021

Application of Topological Data Analysis for Biomechanical Dataset

by

Sangman Jung

Advised by

Kyungsoo Kim

Submitted to the Department of Mathematics
and the Faculty of the Graduate School of
Kyung Hee University in partial fulfillment
of the requirements for degree of
Master of Science

Dissertation Committee:

Sunmi Lee (Chairman)

Jeongsan Kim

Kyungsoo Kim

Abstract

Application of topological data analysis for biomechanical dataset

By Sangman Jung

Master of Science in Mathematics

Graduate School of Kyung Hee University

Advised by Kyungsoo Kim

In this work, we introduce an application of topological data analysis, which is a new framework of data analysis and a combination of topology and computation. Topological data analysis employs the persistence concept with the homology theory in algebraic topology so that it can be more efficient to detect hidden topological features of the data than the other existing data analysis methods. This method is often useful to the data analysis process with conventional data analysis methods such as hierarchical clustering, k-means clustering, multidimensional scaling. We used these conventional methods to analyze the internal relationship of variables. We also developed the GUI (Graphical User Interface) program to implement our process in this paper. We analyzed joint kinematics in biomechanics for 9 subjects with 31 variables as an application case. Our finding is that the joint angles of right body parts cause the difference between subjects, and also we found that subject 4 and subject 7, and subject 8 and 9 have a similar topological structure for each other.

Keywords

Topological data analysis, Hierarchical clustering, Multidimensional scaling, GUI program, Biomechanical dataset

Table of Contents

Abstract	i
List of Figures	iv
List of Tables	vii
Chapter 1. Introduction	1
Chapter 2. Analysis Methods	4
2.1 Conventional data analysis	4
2.2 Topological data analysis	10
2.2.1 Persistent homology	10
2.2.2 Barcode and persistence diagram	15
2.2.3 Bottleneck distance	16
Chapter 3. Development of GUI Program	19
3.1 Introduction	19
3.1.1 Basic structure	19
3.2 Details of the analysis tools	22
3.2.1 Data Loading Tab	22
3.2.1.1 Browsing and loading a dataset	22
3.2.1.2 Usage of Viewer and Results	24
3.2.2 Data Analysis Tab	27
3.2.2.1 Hierarchical Clustering Toolbox Tab	27
3.2.2.2 Multidimensional Scaling Toolbox Tab	31
3.2.2.3 Topological Data Analysis Toolbox Tab	36

Chapter 4. Application of Topological Data Analysis	40
4.1 Biomechanical dataset	40
4.2 Results	41
4.3 Discussion	50
Chapter 5. Conclusion	51
References	52



List of Figures

Figure 2.1. Example of two representations from a point clouds as the (dis)similarity matrix $d_C \in \mathbb{R}^{3 \times 3}$. (a) The graph representation of d_C . x_1, x_2, x_3 are nodes and the straight lines between nodes are the edges with edge weights. (b) The heatmap representation of d_C . Each cell of the heatmap corresponds to each value of d_C as the edge weights.	4
Figure 2.2. Example of the relationship between the barcode for β_0 and the single linkage dendrogram from a point clouds as the (dis)similarity matrix $d_C \in \mathbb{R}^{12 \times 12}$	5
Figure 2.3. Example of the relationship between the single linkage dendrogram and its linkage matrix. In the middle of the figure, the upper of two heatmaps is for the 0th-Betti number, and the lower is the single linkage matrix.	6
Figure 2.4. Example of the construction of the single linkage matrix $d_S \in \mathbb{R}^{5 \times 5}$. The upper of this figure represents the calculation process of the single linkage method in the matrix. The middle is the direct calculation of the single linkage method. The bottom represents the heatmap of d_S	7
Figure 2.5. The general workflow of the computation of MDS.	9
Figure 2.6. Example of k -simplex for $k = 0, 1, 2, 3$	11
Figure 2.7. Example of the Čech complex (a) and Vietoris-Rips complex (b).	12
Figure 2.8. Example of k th Betti number β_k when $k = 0, 1, 2$	14
Figure 2.9. Example of the barcode and persistence diagram for the filtration of given point cloud data. (a) represents the filtration process of the point clouds. (b) represents the barcode for β_k when $k = 0, 1, 2$. (c) is the persistence diagram corresponding to the barcode for β_k	16
Figure 2.10. Example of the computation of the bottleneck distance from the barcode for β_1 . (a) Two barcodes for β_1 and the description of the point of birth and death in their PDs. (b) Overlapped PD and the correspondence between the points for two PDs. (c) The magnified picture of (b).	18
Figure 3.1. GUI_AppTDA execution screen.	20
Figure 3.2. The flowchart of GUI_AppTDA.	21

Figure 3.3. Browsing the dataset.	22
Figure 3.4. Usage of the ‘Data Matrix Editor’.	23
Figure 3.5. The process of loading the dataset. In this figure, the upper is the screen for the usage of loading the data with the checkbox ‘scaling and centering’. The lower is the screen for the button ‘Initialize’.	24
Figure 3.6. Usage of the button ‘view’ in the ‘Viewer’.	25
Figure 3.7. Usage of the button ‘save’ in the ‘Viewer’.	26
Figure 3.8. Usage of the heatmap tool. In this figure, the left is the location of the heatmap tool, and the right is the result when ‘Colormap’ is ‘pink’, and ‘Distances’ is a correlation-based distance matrix ($31 \times 31 \times 9$).	27
Figure 3.9. The colormaps in MATLAB.	26
Figure 3.10. The result of the single linkage dendrogram when ‘Colormap’ is ‘hot’, and ‘Distances’ is a correlation-based distance matrix ($31 \times 31 \times 9$).	28
Figure 3.11. The ‘Barcode Curve (Betti-0)’ panel (top), and its result (bottom), when ‘Colormap’ is ‘jet’, ‘Point cloud’ is correlation-based distance matrix, ‘Max Filtration’ is 1.	29
Figure 3.12. Usage of the ‘Barcode Area Calculator’. The left is the figure that the calculator is activated, and the right is the results of the calculator as clicking the ‘view’ button.	30
Figure 3.13. The Shepard plot tool of MDS Toolbox Tab.	31
Figure 3.14. Two Shepard plot results for each page of the correlation-based distance array. (a) is the result of the Shepard plot for 2D, using the loss function ‘metricstress’, and (b) is the result of the Shepard plot for 3D, using the loss function ‘stress’.	32
Figure 3.15. The scree plot for 3D MDS for the criterion ‘metricstress’. In this case, $k = 9$	33
Figure 3.16. The results of the MDS tool.	34
Figure 3.17. The MDS results at 3D or more.	35
Figure 3.18. The barcode panel in TDA Toolbox Tab.	36
Figure 3.19. The β_k barcode plot for $k = 0$ (the upper), $k = 1$ (the middle), $k = 2$ (the bottom).	37
Figure 3.20. The persistence diagram for β_k , where $k = 0$ (the left), $k = 1$ (the center), $k = 2$ (the right).	38

Figure 3.21. The computation result of the bottleneck distance.	38
Figure 3.22. The warning message for computing the bottleneck distance.	39
 Figure 4.1. The heatmap result of the correlation-based distance matrix d_C for 9 subjects.	41
Figure 4.2. The curve representation of the barcode (β_0) for each subject. The vertical and horizontal axes represent the number of connected components and filtration values.	42
Figure 4.3. The heatmap result of the single linkage matrix d_S for 9 subjects.	43
Figure 4.4. The single linkage dendrogram for 9 subjects. The vertical axis represents 31 connected components as FE (10), AD (10), IE (10), Xf (X-factor). The horizontal axis represents the filtration value.	44
Figure 4.5. The Shepard plot of the Gromov-Hausdorff distance for the 3D MDS. ‘D’, ‘C’, ‘P’ means ‘dimension’, ‘criterion’, and the ‘Pearson correlation coefficient’.	47
Figure 4.6. The scree plot of three distances for the 3D MDS. (a) is the bottleneck distance for β_0 , (b) is the bottleneck distance for β_1 , and (c) is the Gromov-Hausdorff distance. We set the optimal k as $k_{BN0} = 3$, $k_{BN1} = 3$, $k_{GH} = 3$, respectively.	47
Figure 4.7. The k-means clustering results of 3D MDS for the bottleneck distance and the Gromov-Hausdorff distance.	49

List of Tables

Table 4.1. The area and slope of the curve of the barcode (β_0) for 9 subjects. The row ‘End_F’ is the end of the filtration which means when all connected components are merged	42
Table 4.2. The Bottleneck distance matrix (β_0) for 9 subjects	45
Table 4.3. The Bottleneck distance matrix (β_1) for 9 subjects	45
Table 4.4. The Gromov-Hausdorff distance matrix for 9 subjects	46



Chapter 1. Introduction

As various types of data can be obtained in many fields through rapid technological advancement, interest, and importance in data analysis are increasing. Accordingly, the various analysis methodology for a given dataset, such as from well-known traditional statistical methods to machine learning [2] and deep learning [3] that recently attracting attention is also studied. These analysis methods may be used singly, but more efficient analysis results are obtained when they are overlapped and applied according to the appropriate situation [4]. Nevertheless, as the complexity of the data increases, it is becoming more difficult to qualitatively analyze the data and identify the characteristics of the data. Topological data analysis (TDA) is one of the analysis methodologies that can be an alternative to this problem and is a subject of relatively recent research [5]. TDA only pays attention to the nature of data, that is, the invariability of the structure, which corresponds to the concept of *homeomorphic* in topology. This concept is to examine whether the topological structure of two topological spaces is the same. Intuitively, the concept of homeomorphic is interested in whether one of the two is transformed without making a hole or tearing it, and whether it becomes the same as the other. Based on this concept, TDA extracts the topological features of the data and compares multiple data qualitatively.

The general workflow of TDA is reduced to the steps of defining a topological space of data, calculating the topology of the topological space, and then summarizing and visualizing the results. Therefore, it is necessary to first define the topological space of the data. The definition of the topological space itself encompasses an abstract arbitrary object, but it can be easily defined by representing the data as a metric space [6]. In detail, calculating a pairwise distance for two elements of the data represented by a matrix or array is used [7]. In TDA, the topological space created in this way is called a *point clouds* [8]. The main technique for computing the topological structure uses *persistent homology* [9], which calculates homology for all topological structures that a given point cloud can have. Intuitively, if there is a track that has information about all the topologies of the data to be analyzed, this method continuously monitors the track and calculates the homology until the point where the topological structure no longer changes. For this reason, it is named ‘persistent’.

To summarize and visualize the results, two major tools in TDA, the *barcode*, and the *persistence diagram* (PD), are used. The barcode is a figure represented by drawing a horizontal bar between the time point at which the topological feature appears and the time point at which it disappears, through continuous homology calculation from the point cloud. The PD is a figure shown on a two-dimensional plane by pairing the point when the topological feature appears and the point when the topological feature disappears. In general, the point at which this topological feature appears is called 'birth', and the point at which it disappears is called 'death'. Also, the interval for these points is called 'filtration', and the value of the interval is called 'filtration value'. Apart from the problem of obtaining the topological characteristics of one data, it is also necessary to compare the topological features of two or more data. This work mainly uses the *bottleneck distance* (BN distance) which is the distance measure for calculating the difference between the pair of persistence diagrams of the two data. [10].

Meanwhile, various software for calculating the persistent homology has been developed. Typically, it can be calculated using JavaPlex [11] in MATLAB, and GUDHI is used in python [12, 13]. For the case of R, the package 'TDA' is developed [14]. The introduction to the overall software packages of TDA is summarized in [15].

In many fields, the papers for applications of persistent homology are published [16, 17]. As an example, H. Lee [18, 19] proposed another alternative to find the optimal threshold by switching from the problem for finding a specific threshold in the traditional network analysis to the problem of TDA, in the human brain connection data of three groups measured by FDG-PET. Besides, they reconstructed the barcode for the connectivity of each node in the brain network to the single linkage dendrogram (SLD) in hierarchical clustering analysis [20] and compared the quantified results of the dendrograms of each group using the *Gromov-Hausdorff distance* (GH distance) [21] which is the distance measure to calculate the difference between two geometrical objects.

Multidimensional scaling (MDS) is widely used as one of the methods of visualizing the results that quantified the differences between multiple objects. More specifically, MDS visualizes the result through the dimensionality reduction of a distance matrix or a (dis)similarity matrix [22]. As an example, there are papers for visualizing the dissimilarity between objects of

handwriting data using the bottleneck distance as a distance matrix [23] and investigating the relationship between orthodontic treatment effects by calculating the Wasserstein distance which is a generalized case of the BN distance [24]. In addition, MDS is applied in various types of data based on TDA [25, 26, 27].

In this study, we briefly review the well-known techniques of TDA mentioned above and apply them to the actual dataset. In Section 2.1, we introduce SLD, GH distance, and MDS as analysis methods to be used with TDA. In Section 2.2, persistent homology, barcode and PD, and BN distance are described. In Chapter 3, we introduce the software that can compute the results of this paper. More specifically, a graphical user interface (GUI) developed in a MATLAB-based environment will be introduced, and a manual to easily implement the algorithms introduced in this paper will be presented. In Chapter 4, the dataset that measured the joint angle of the swing motion of 9 golf players was analyzed.



Chapter 2. Analysis Methods

2.1 Conventional data analysis

We briefly explain the conventional analysis methods to be used with TDA. More information on the TDA can be found in Section 2.2. First, we introduce two representations to visualize the point clouds, such as the heatmap representation and the graph representation. In this paper, we only use the heatmap representation for visualization, but the graph-theoretical concept often useful to understand the shape of a topological space. See Figure 2.1.

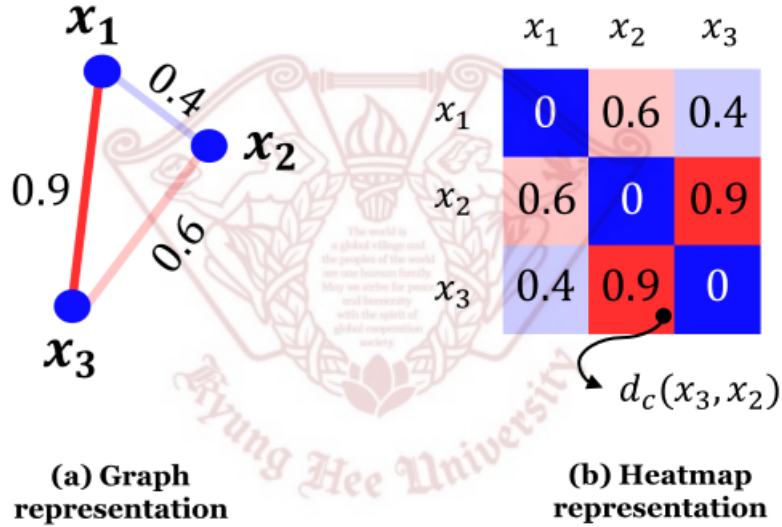


Figure 2.1. Example of two representations from a point clouds as the (dis)similarity matrix $d_C \in \mathbb{R}^{3 \times 3}$. (a) The graph representation of d_C . x_1, x_2, x_3 are nodes and the straight lines between nodes are the edges with edge weights. (b) The heatmap representation of d_C . Each cell of the heatmap corresponds to each value of d_C as the edge weights.

In Figure 2.1, let $d_C \in \mathbb{R}^{3 \times 3}$ is the (dis)similarity matrix which is the distance matrix such that the matrix is symmetric and the diagonal members are defined as zero, then each cell of the heatmap correspond to each member of d_C . The color of the cell in the heatmap determined by the value of the member of d_C .

In many clustering analysis methods, we used hierarchical clustering which finds the hierarchy of the clusters. Because it is easy to give a hierarchical structure to the point cloud data since this method uses the (dis)similarity matrix as input. Also, this method at the specific linkage criterion grant to the barcode for β_0 the hierarchy. That linkage criterion is the single linkage method. The details of this linkage method are in [20]. Using the single linkage method, we obtain the dendrogram (or the single linkage dendrogram) which represents the hierarchical relationship between objects, like a tree. In this paper, the dendrogram is important to grant the geometrical information to the barcode. More specifically, according to [19], the SLD is equivalent to the barcode for β_0 , and the fact in [19] adds hierarchical features to the barcode. In other words, the Vietoris-Rips filtration for β_0 is equivalent to the construction of the SLD. The relationship between the dendrogram and the barcode is presented in Figure 2.2.

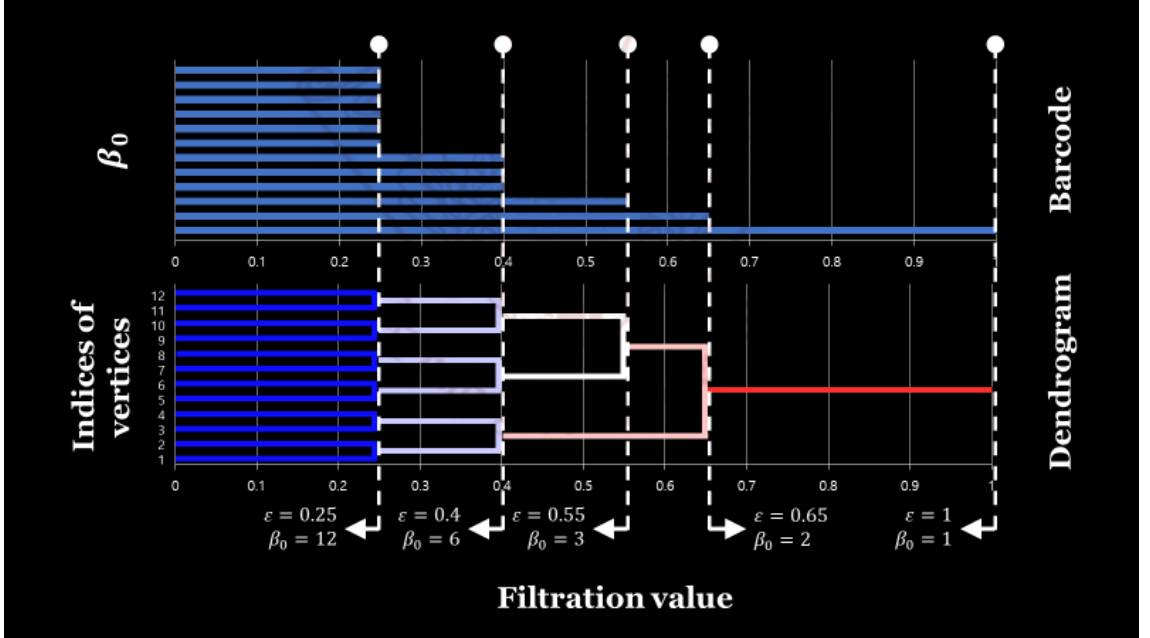


Figure 2.2. Example of the relationship between the barcode for β_0 and the single linkage dendrogram from a point clouds as the (dis)similarity matrix $d_C \in \mathbb{R}^{12 \times 12}$.

In Figure 2.2, we set $d_C \in \mathbb{R}^{12 \times 12}$, and draw the barcode and the dendrogram. The vertical axis represents the indices of vertices (nodes) or variables, and the horizontal axis represents the height

as filtration value. The practical computation of the SLD needs the minimum spanning tree which is the algorithm to find the shortest path in the spanning tree.

Also, we can save the information of the SLD to a matrix form. This called the single linkage matrix d_S . The construction of this matrix is described in [20]. The small example of $d_S \in \mathbb{R}^{5 \times 5}$ is as shown in Figure 2.4. In this figure, d_S^i is the single linkage matrix for step i which is a hierarchy. The relationship between the SLD and its linkage matrix presents in Figure 2.3.

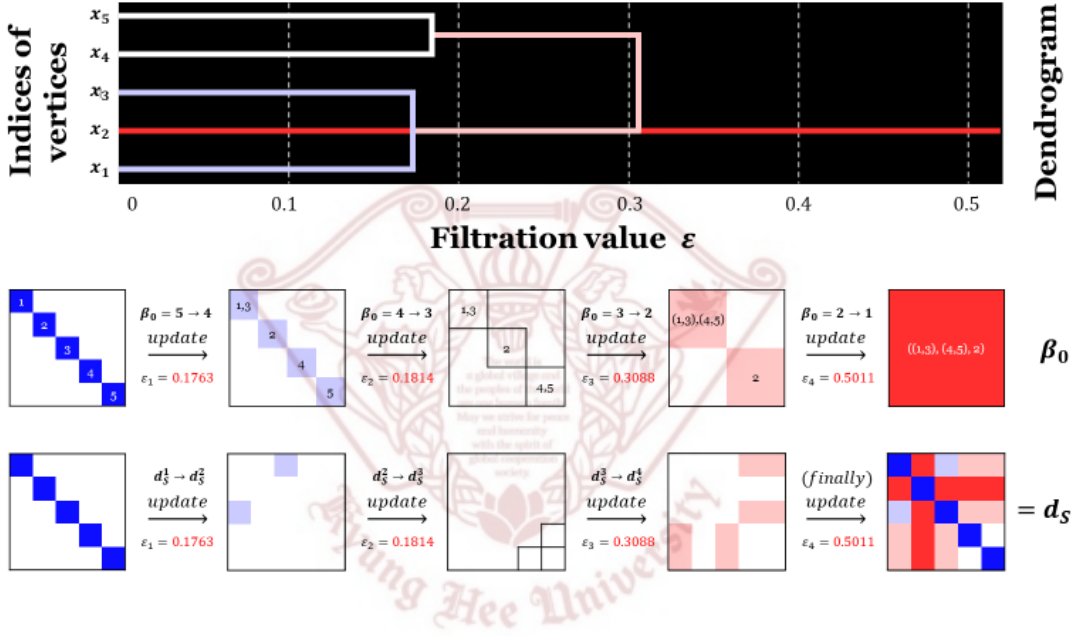
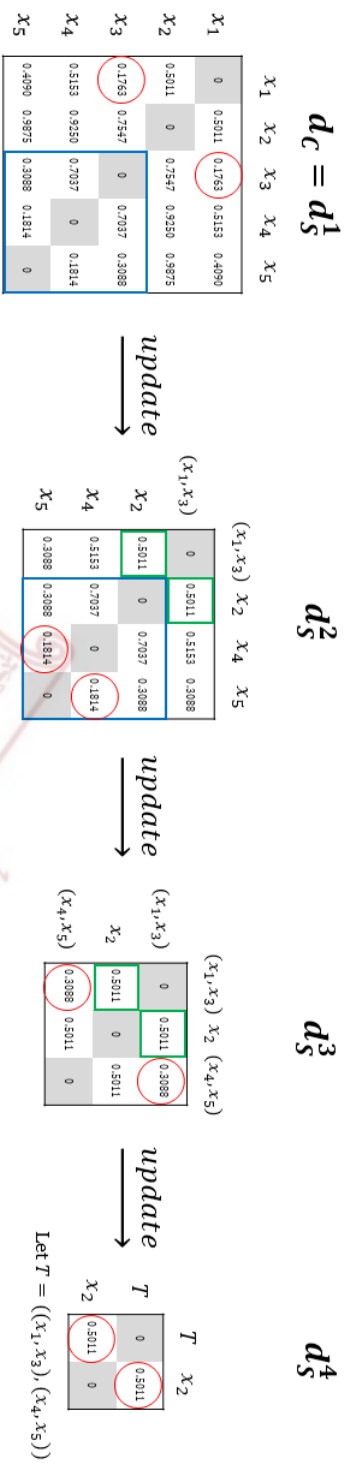


Figure 2.3. Example of the relationship between the single linkage dendrogram and its linkage matrix. In the middle of the figure, the upper of two heatmaps is for the 0th-Betti number, and the lower is the single linkage matrix.

The single linkage matrix is to compute the GH distance [21]. This distance is the measure of dissimilarity between shapes in Euclidean space. The main feature of the GH distance is that the shapes should be regarded as metric space. This measure provides a natural tool for studying the perturbation of the inputs and outputs of hierarchical clustering methods [20]. The definition of the GH distance between two metric spaces with a single linkage distance is as follows.



$d_1^1(C_1, C_2) = \min_{x_i \in C_1, x_j \in C_2} d_1(x_i, x_j) = 0.1753 \leftarrow$ let a filtration value (height)

Thus, we cluster (x_1, x_3) , then we can update a new d_1^2 :

Updated clusters (vertices): $((x_1, x_3)) \in C_1, (x_2, x_4, x_5) \in C_2$

Updated edges:

$d_1^2((x_1, x_3), x_2) = \min(d_1^1(x_1, x_2), d_1^1(x_3, x_2)) = (0.5011, 0.7547) = 0.5011$

$d_1^2((x_1, x_3), x_4) = \min(d_1^1(x_1, x_4), d_1^1(x_3, x_4)) = (0.5153, 0.7037) = 0.5153$

$d_1^2((x_1, x_3), x_5) = \min(d_1^1(x_1, x_5), d_1^1(x_3, x_5)) = (0.4090, 0.3088) = 0.3088$

$d_1^2(C_1, C_2) = \min_{x_i \in C_1, x_j \in C_2} d_1^2(x_i, x_j) = 0.1814 \leftarrow$ height = $d_1^2 - d_1^1 = 0.0051$

Thus, we cluster (x_4, x_5) , then we can update a new d_1^3 :

Updated clusters (vertices): $((x_4, x_5)) \in C_1, (x_2, (x_1, x_3)) \in C_2$

Updated edges:

$d_1^3((x_4, x_5), x_1) = \min(d_1^2(x_4, x_1), d_1^2(x_5, x_1)) = (0.5011, 0.9875) = 0.5011$

$d_1^3((x_4, x_5), x_2) = \min(d_1^2(x_4, x_2), d_1^2(x_5, x_2)) = (0.1814, 0.3088) = 0.3088$

$d_1^3(C_1, C_2) = \min_{x_i \in C_1, x_j \in C_2} d_1^3(x_i, x_j) = 0.3088 \leftarrow$ height = $d_1^3 - d_1^2 = 0.1274$

Thus, we cluster $((x_4, x_5), (x_1, x_3))$, then we can update a new d_1^4 :

Updated clusters (vertices): $((x_1, x_3), (x_4, x_5)) \in C_1, (x_2) \in C_2$

Updated edges:

$d_1^4(((x_1, x_3), (x_4, x_5)), x_2) = \min(d_1^3((x_1, x_3), x_2), d_1^3((x_4, x_5), x_2)) = 0.5011$

$d_1^4(C_1, C_2) = \min_{x_i \in C_1, x_j \in C_2} d_1^4(x_i, x_j) = 0.5011 \leftarrow$ height = $d_1^4 - d_1^3 = 0.1933$



Figure 2.4. Example of the construction of the single linkage matrix $d_s \in \mathbb{R}^{5 \times 5}$. The upper of this figure represents the calculation process of the single linkage method in matrix. The middle is the direct calculation of the single linkage method.

The bottom represents the heatmap of d_s .

Definition 2.1. (Gromov-Hausdorff distance [37]) Given two metric spaces (X, d_{S_X}) , (Y, d_{S_Y}) , the Gromov-Hausdorff distance between X and Y is defined as

$$d_{GH}(X, Y) = \frac{1}{2} \max_{\forall i, j} |d_{S_X}(x_i, x_j) - d_{S_Y}(y_i, y_j)|.$$

To visualize the (dis)similarity, we used MDS which is one of the dimensionality reduction methods. This method includes classical MDS, metric MDS, and non-metric MDS. In particular, the classical MDS is the same as the principal component analysis. Each MDS method has a loss function. The loss function of classical MDS is called ‘strain’, and the loss functions of metric MDS and non-metric MDS are called ‘stress’. MDS minimize these loss function to obtain the approximately accurate results in a lower dimension. The details of the loss functions and algorithms are in [22, 38]. We used the loss functions of MATLAB that are ‘strain’ for classical MDS, ‘metricstress’ for metric MDS, and ‘stress’ for non-metric MDS. ‘sstress’ and ‘metricsstress’ are the squared stress for non-metric MDS, metric MDS, respectively. The loss function ‘sammon’ [39] in metric MDS is also used. The general workflow of the computation of MDS is as shown in Figure 2.5. Let $D \in \mathbb{R}^{p \times p}$ is the dissimilarity matrix to be analyzed, then each column and each row represents the variable V_i . Then, MDS provides an insight into the relationship between V_i , $i = 1, 2, \dots, p$, by approximating each member d_{ij} of D to the coordinates in a lower dimension ($q < p$) using appropriate loss function. By this approximation, we obtain the $p \times q$ matrix, in other words, the q -dimensional coordinates for the input D . This coordinates matrix can represent the 2D or 3D scatter plot, and the points of the plot are interpreted as V_i ’s.

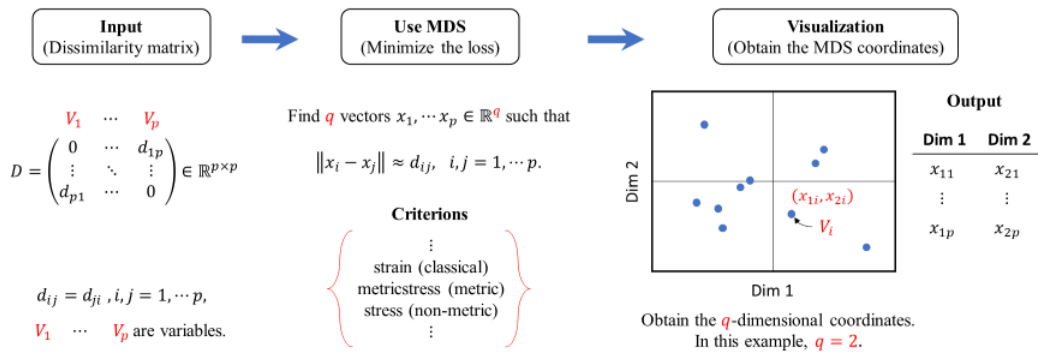


Figure 2.5. The general workflow of the computation of MDS.



2.2 Topological data analysis

We briefly introduce the main techniques of TDA. In Section 2.2.1, we describe the contents necessary to understand persistent homology. In Section 2.2.2, we focus on the intuitive example of the barcode and the PD and interpretation of its figures. In the last section, we explain the definition and calculation of the BN distance comparing two PDs.

2.2.1 Persistent homology

Persistent homology is a methodology based on the homology theory, which is dealt with in algebraic topology and extracts the topological features of a given data through continuous homology calculations. This methodology mainly uses a method of gluing a generalized triangle called *simplex* to approximate it to the original shape of the point cloud data. Then, we compute the homology of the approximated shape through finitely many simplices. To get the approximate shape from the point cloud, we first need to define a simplex. The definition is as follows.

Definition 2.2. (k -simplex [28]) A k -simplex is the convex hull of $k+1$ linearly independent points $S = \{v_0, v_1, \dots, v_k\}$. The points in S are the vertices of the simplex.

Note that the *convex hull* $S = \{p_0, p_1, \dots, p_k\} \subseteq \mathbb{R}^d$ is the set of all convex combination which is a linear combination $x = \sum_{i=0}^k \lambda_i p_i$, for some $\lambda_i \in \mathbb{R}^+ \cup \{0\}$, and $\sum_{i=0}^k \lambda_i = 1$. Also, note that a k -simplex is a k -dimensional subspace of \mathbb{R}^d . If $S = \{v_0, v_1, v_2, v_3\}$, 0-simplex $\{v_0\}$ is a vertex, 1-simplex $\{v_0, v_1\}$ is an edge, 2-simplex $\{v_0, v_1, v_2\}$ is a triangle, and 3-simplex $\{v_0, v_1, v_2, v_3\}$ is tetrahedron. An example of intuitively understanding simplex is shown in Figure 2.6. The topological space is approximated by the combination of finitely many simplices. The set of these finitely many simplices satisfy the specific condition is called *simplicial complex*. The definition is as follows.

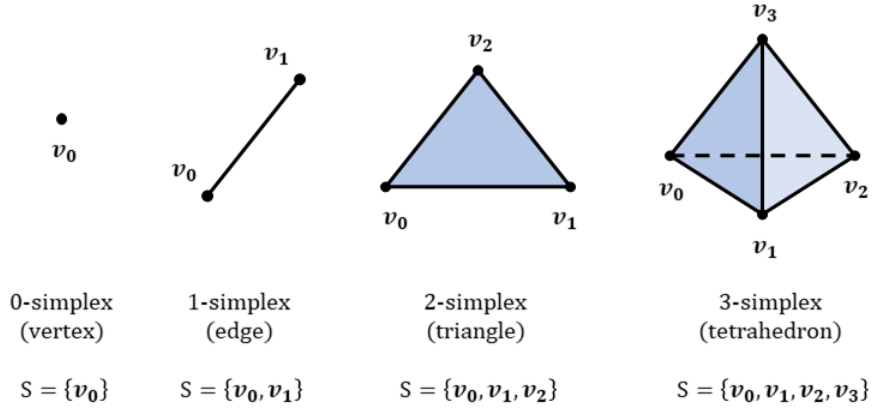


Figure 2.6. Example of k -simplex for $k = 0, 1, 2, 3$.

Definition 2.3. (Simplicial complex [28]) A *simplicial complex* K is a finite set of simplices such that (1) $\sigma \in K, \tau \leq \sigma \Rightarrow \tau \in K$ and (2) $\sigma, \sigma' \in K \Rightarrow \sigma \cap \sigma' \leq \sigma, \sigma'$.

In this definition, σ is a k -simplex defined by $S = \{p_0, p_1, \dots, p_k\}$, τ is a *face* of σ . Note that τ is a simplex defined by $T \subseteq S$. There are many construction methods of a simplicial complex from the dataset. Among them, the most commonly used methods can be divided into 4 types: Čech complex, Vietoris-Rips complex, alpha complex, and witness complex. In this paper, we construct a simplicial complex using the Vietoris-Rips complex, which is the only practical complex for analyzing datasets in higher dimensions [29]. This is because, the Čech complex is a reasonable theoretical method, but its computation is difficult. On the other hand, the Vietoris-Rips complex is much easier to compute the persistent homology and approximate the Čech complex. For $S = \{p_0, p_1, \dots, p_k\} \subseteq \mathbb{R}^d$, the definition of the Vietoris-Rips complex is as follows.

Definition 2.4. (Vietoris-Rips complex [29]) The *Vietoris-Rips complex* (VR complex) $\mathcal{V}_\varepsilon(S)$ of S at scale ε is $\mathcal{V}_\varepsilon(S) = \{\sigma \subseteq S \mid d(u, v) \leq \varepsilon, \forall u \neq v \in \sigma\}$, where d is the Euclidean metric.

Note that d can be a pairwise distance on arbitrary finite metric space. In addition, the VR complex has the following containment relation with the Čech complex.

Theorem 2.5. (Vietoris-Rips Lemma in [30]) Let $\check{C}_\varepsilon(S)$ is the Čech complex and $\mathcal{V}_\varepsilon(S)$ is the Vietoris-Rips complex. For any $\varepsilon \geq 0$, we have $\check{C}_\varepsilon(S) \subseteq \mathcal{V}_\varepsilon(S) \subseteq C_{\sqrt{2}\varepsilon}(S)$.

The definition of the Čech complex can be found at [30, 31]. Theorem 2.4. allows the VR complex construction to be used instead of the Čech complex construction. The example of the Čech complex and Vietoris-Rips complex presented in Figure 2.7

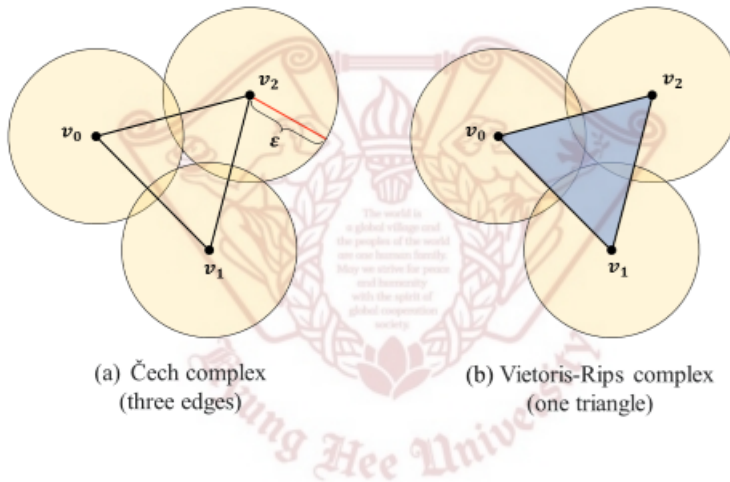


Figure 2.7. Example of the Čech complex (a) and Vietoris-Rips complex (b).

The homology groups are defined on the chain complex which represents the relationships between the cycles and boundaries of simplicial complexes in various dimensions of a topological space.

Definition 2.6. (Chain group [28]) The k th chain group of a simplicial complex K is $\langle C_k(K), + \rangle$, the free Abelian group on the oriented k -simplices, where $[\sigma] = -[\tau]$ if $\sigma = \tau$ and σ and τ have different orientations. An element of $C_k(K)$ is a k -chain, $\sum_q n_q [\sigma_q]$, $n_q \in \mathbb{Z}$, $\sigma_q \in K$.

Definition 2.7. (Boundary homomorphism [28, 30]) Let K be a simplicial complex and $\sigma \in K$, $\sigma = [v_0, v_1, \dots, v_k]$. The *boundary homomorphism* $\partial_k : C_k(K) \rightarrow C_{k-1}(K)$ is

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

where \hat{v}_i indicates that v_i is deleted from the sequence.

Note that $\partial_{k-1}\partial_k = 0$ for all k . Using the boundary homomorphism, we obtain the following sequence of homomorphisms for k -dimensional complex K .

$$0 \rightarrow C_k \xrightarrow{\partial_k} C_{k-1} \xrightarrow{\partial_{k-1}} \dots \rightarrow C_1 \xrightarrow{\partial_1} C_0 \xrightarrow{\partial_0} 0$$

This sequence is called a *chain complex* [32]. Note that the image of ∂_{k+1} , $\text{Im}\partial_{k+1}$ and the kernel of ∂_k , $\text{Ker}\partial_k$ are free Abelian normal subgroups of C_k , and $\text{Im}\partial_{k+1}$ is a normal subgroup of $\text{Ker}\partial_k$ [28]. Also, note that $\text{Ker}\partial_k$ is called *kth cycle group* and its element is called a *k-cycle*, and $\text{Im}\partial_{k+1}$ is called the *kth boundary group* and its element is called a *k-boundary*. Now we can define the *kth simplicial homology group* of the chain complex as follows.

Definition 2.8. (*kth simplicial homology group* [32]) The *kth simplicial homology group* of the chain complex $C_k(K)$ for a simplicial complex K is a quotient group $H_k = \text{Ker}\partial_k / \text{Im}\partial_{k+1}$.

There is an important indicator of information about the *kth homology group's k-dimensional topological feature*. Such a topological invariant is called *kth Betti number*. The definition is as follows.

Definition 2.9. (The *kth Betti number* [30]) The *kth Betti number* of $H_k = \text{Ker}\partial_k / \text{Im}\partial_{k+1}$ is the rank of H_k , denoted by $\beta_k = \text{rank } H_k$.

Informally, the k th Betti number is the number of k -dimensional voids in the simplicial complex [33]. For example, β_0 is the number of connected components (connectivity) and β_1 is the number of 1-dimensional holes or loops and β_2 is the number of enclosed solid voids (2-dimensional voids) [34]. Figure 2.8 shows an example of the k th Betti number when $k = 0, 1, 2$.

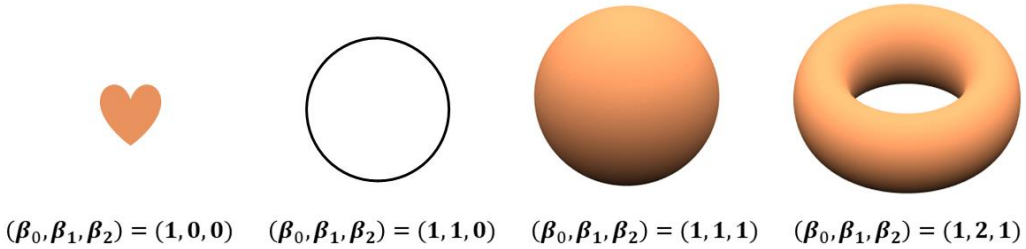


Figure 2.8. Example of k th Betti number β_k when $k = 0, 1, 2$.

From the far left in Figure 2.8, the shape of a heart is regarded as a point and becomes a connected component. Thus $\beta_0 = 1$ and all other cases are 0. The circle at the second location in Figure 2.8 is the 1-dimensional loop by itself. Therefore, $\beta_1 = 1$ and $\beta_0 = 1$, since the loop is also the kind of one connected component. The sphere at the third location is the same as the case of the circle, but it additionally has the 2-dimensional void, hence $\beta_2 = 1$. The torus at the last has two independent 1-dimensional loops, so $\beta_1 = 2$.

To define the persistent homology, the concept ‘filtration’ is important. This concept is for the continuous calculation of the simplicial homology. The definition of filtration is as follows.

Definition 2.10. (Filtration [35]) A *filtration* of a complex K is a nested subsequence of complexes $\emptyset = K^0 \subseteq K^1 \subseteq \dots \subseteq K^m = K$. For generality, we let $K^i = K^m$ for all $i \geq m$. In this situation, K is called a *filtered complex*.

For every $i \leq j$, we define the inclusion map $K_i \hookrightarrow K_j$ induces a homomorphism

$$f_p^{i,j} : H_p(K_i) \rightarrow H_p(K_j)$$

for each dimension p . Then, the filtration corresponds to a sequence of homology groups connected by homomorphisms, $0 = H_p(K_0) \rightarrow H_p(K_1) \rightarrow \dots \rightarrow H_p(K_m) = H_p(K)$ for each dimension p [30]. Then, the definition of the p^{th} persistent homology groups are as follows.

Definition 2.11. (The p th persistent homology groups [30]) The p th persistent homology groups are the images of the homomorphisms induced by inclusion, $H_p^{i,j} = \text{Im} f_p^{i,j}$, for $0 \leq i \leq j \leq m$. The corresponding p th persistent Betti numbers are the ranks of these groups, $\beta_p^{i,j} = \text{rank} H_p^{i,j}$.

Note that $H_p^{i,i} = H_p(K_i)$. By definition 2.10 and 2.11, we can define the p th persistent homology groups along with the Vietoris-Rips filtration, which is the filtration of VR complex $\mathcal{V}_\varepsilon(K^i)$ of simplicial complex K^i at scale $\varepsilon \geq 0$.

2.2.2 Barcode and persistence diagram

The rigorous definition of the PD and the barcode is found at [10, 30]. In this Subsection, we only describe the practical aspect of these methods. The PD is the diagram for visualizing the collection of persistent Betti numbers by drawing points in two dimensions [30]. In other words, the PD is the scatter plot of the point set contains the points which are paired up with the filtration value of the appeared topological feature and the filtration value of the disappeared topological feature [36]. The points of the PD correspond to the filtration values of the horizontal bars of the barcode. The example of the barcode and the PD is shown in Figure 2.9.

For arbitrary $\varepsilon \geq 0$, first, we construct the complexes for given point clouds as (a). Informally, this process is often expressed to making the complexes as the ε -ball of each point increasing under the specific complex construction method. In this situation, we compute the persistent homology for each ε , and obtain the k th Betti number to represent the barcode or PD. When the topological feature appears in the filtration, we record it as a bar of the horizontal bar graph. Here, the length of this bar is $|\text{feature death} - \text{feature birth}|$. This representation is the barcode (in (b)). Now, then we can take the values of the feature birth and death and represent a pair of points into

a 2D plane. This representation is the PD (in (c)). The diagonal line of PD is the points at birth = death, we can check that the points nearby the diagonal line are noise.

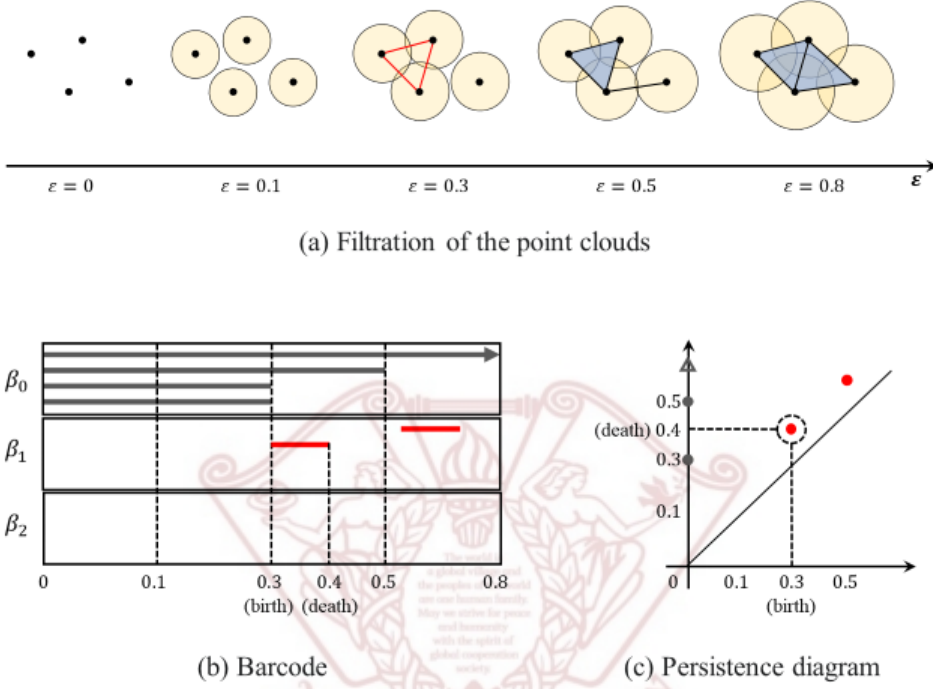


Figure 2.9. Example of the barcode and persistence diagram for the filtration of given point cloud data. (a) represents the filtration process of the point clouds. (b) represents the barcode for β_k when $k = 0, 1, 2$. (c) is the persistence diagram corresponding to the barcode for β_k .

2.2.3 Bottleneck distance

In this Subsection, we explain the definition of the BN distance and its computation. The BN distance is the measure to compare two sets in the same metric space [19]. Let us consider two functions $f, g : \mathbb{R} \rightarrow \mathbb{R}$, and define a bijection $\gamma : Dgm(f) \rightarrow Dgm(g)$ where $Dgm(f)$, $Dgm(g)$ are the persistence diagrams of f, g , respectively. Assume that $m = n$ where $|Dgm(f)| = m$, $|Dgm(g)| = n$. Then, the BN distance of the corresponding two PDs [36] is

$$d_{BN}(Dgm(f), Dgm(g)) = \inf_{\gamma} \sup_{1 \leq i \leq m} \|x_i^f - \gamma(x_i^f)\|_{\infty}$$

where $x_i^f = (B_i^f, D_i^f) \in Dgm(f)$. The infimum is taken over all bijections. If $x_i^g = (B_i^g, D_i^g) = \gamma(x_i^f)$ for some i and j , we can rewrite the formula as

$$d_{BN}(Dgm(f), Dgm(g)) = \min_{\gamma} \max_{1 \leq i \leq m} \|x_i^f - \gamma(x_i^f)\|_{\infty},$$

where

$$\|x_i^f - \gamma(x_i^f)\|_{\infty} = \max\{|B_i^f - B_j^g|, |D_i^f - D_j^g|\}.$$

If we assume $m \neq n$, then there is no bijection between two PDs. Then we generate the auxiliary points as

$$\left(\frac{B_1^f + D_1^f}{2}, \frac{B_1^f + D_1^f}{2}\right), \dots, \left(\frac{B_m^f + D_m^f}{2}, \frac{B_m^f + D_m^f}{2}\right) \text{ and } \left(\frac{B_1^g + D_1^g}{2}, \frac{B_1^g + D_1^g}{2}\right), \dots, \left(\frac{B_m^g + D_m^g}{2}, \frac{B_m^g + D_m^g}{2}\right)$$

that are orthogonal projections to the diagonal line $B = D$ in $Dgm(f)$ and $Dgm(g)$. These points are added to $Dgm(f)$ and $Dgm(g)$, respectively, to make the identical number of points in two PDs. Figure 2.10 shows the example of the computation of the BN distance for β_1 . The important point of the practical computation of the BN distance is finding the optimal γ . This can be solved through the computation of a perfect matching in a bipartite graph to find the infimum of all bijections. Further details of this distance are described in [10].

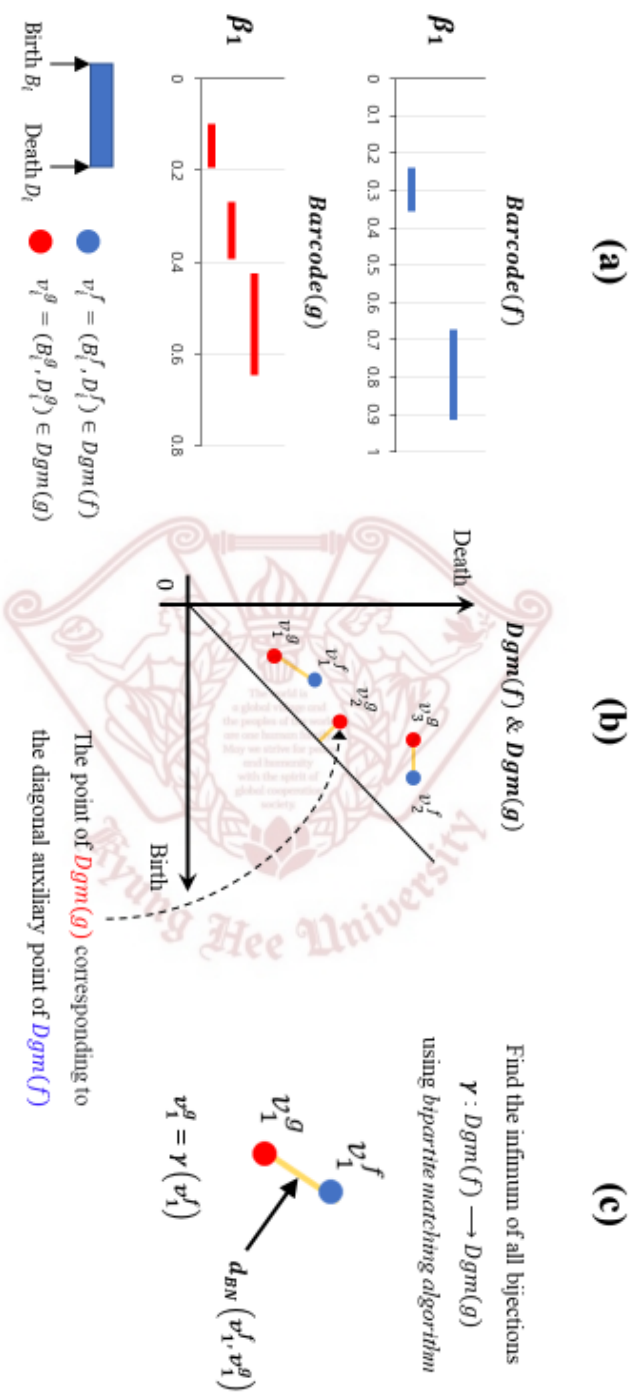



Figure 2.10. Example of the computation of the bottleneck distance from the barcode for β_1 . (a) Two barcodes for β_1 and the description of the point of birth and death in their PDs. (b) Overlapped PD and the correspondence between the points for two PDs. (c) The magnified picture of (b).

Chapter 3. Development of GUI program

3.1 Introduction

The GUI_AppTDA is a MATLAB software for analyzing the data using the method of topological data analysis based on the methods of unsupervised learning. This software is designed to analyze effectively when users are unfamiliar with programming or in an environment where MATLAB is not installed. If the user is familiar with MATLAB programming, an example is attached in Chapter 3 so that you can manipulate various settings more specifically. GUI_AppTDA first calculates the correlation-based distance to be a point cloud and then extracts the features of the data using a hierarchical clustering method, dimensionality reduction, and topological data analysis techniques. In this paper, we introduce a brief overview and practical use of them.

3.1.1 Basic structure

This program has two major platforms. In Figure 3.1, the left part of Figure 3.1 is divided into data loading, which loads and edits data, and the right part is divided into data analysis. See the flowchart in Figure 3.2. Each function of the Toolbox can be executed individually, and the output can also be viewed or saved. There are two ways to run the program: #1. (without MATLAB) Install the program and run GUI_AppTDA.exe. #2. (with MATLAB) Run  GUI_AppTDA.mlapp in the current folder tab or enter GUI_AppTDA in the command window.

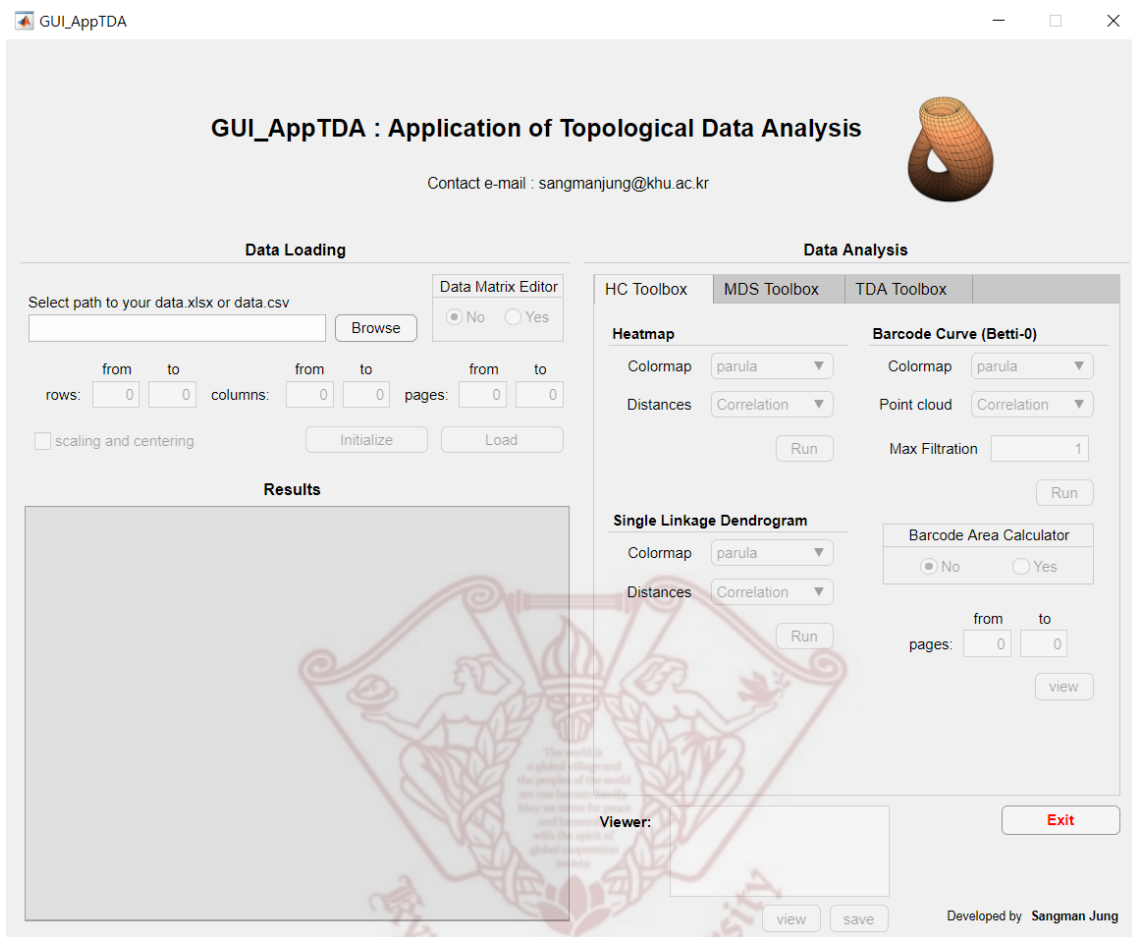


Figure 3.1. GUI_AppTDA execution screen.

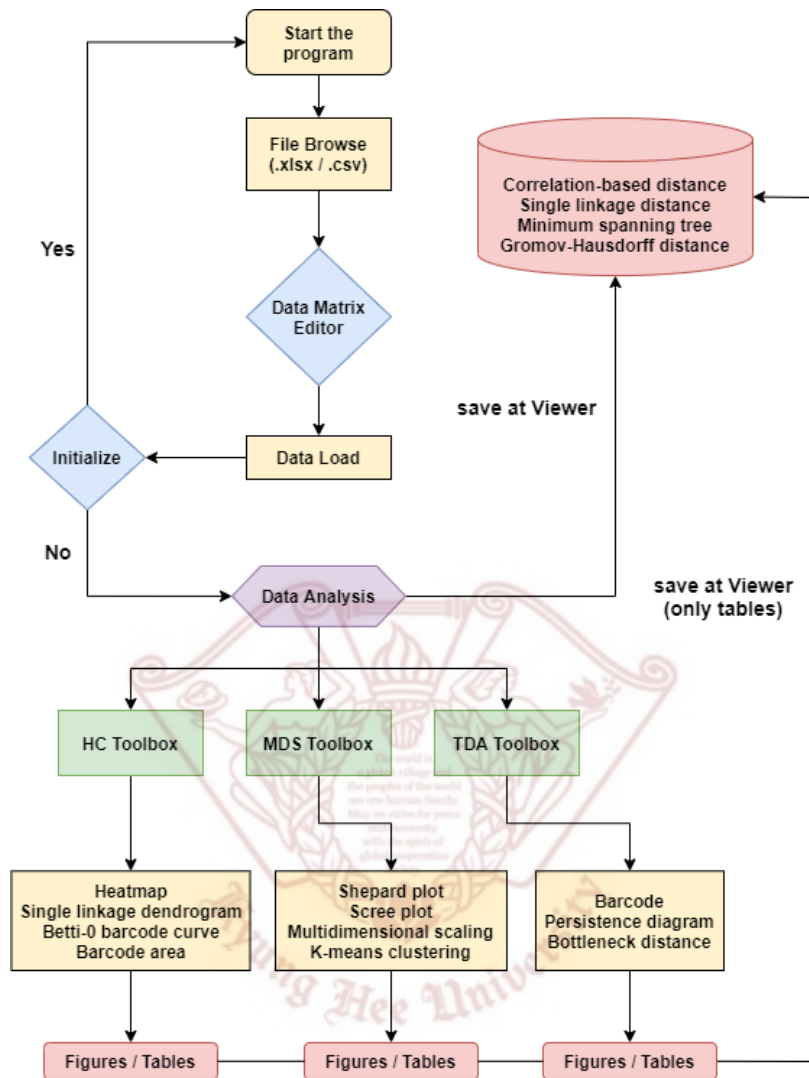


Figure 3.2. The flowchart of GUI_AppTDA.

3.2 Details of the analysis tools

This section deals with the contents of each analysis tool of the program briefly described in the introduction of Chapter 3.

3.2.1 Data Loading Tab

3.2.1.1 Browsing and loading a dataset

Before loading the data, note that only .xlsx and .csv file formats are supported by this program. Also, the columns of the data should be variables or nodes, and the rows of the data should be an observation. If the data is a three-dimensional array, not a matrix, each sheet is regarded as a page of the array. For example, suppose you have data on students' math and English scores for six semesters. Then, each column should be math and English, the rows should be the scores for six semesters, and each page should be the student. Now, click the 'Browse' button and browse the data. See Figure 3.3.

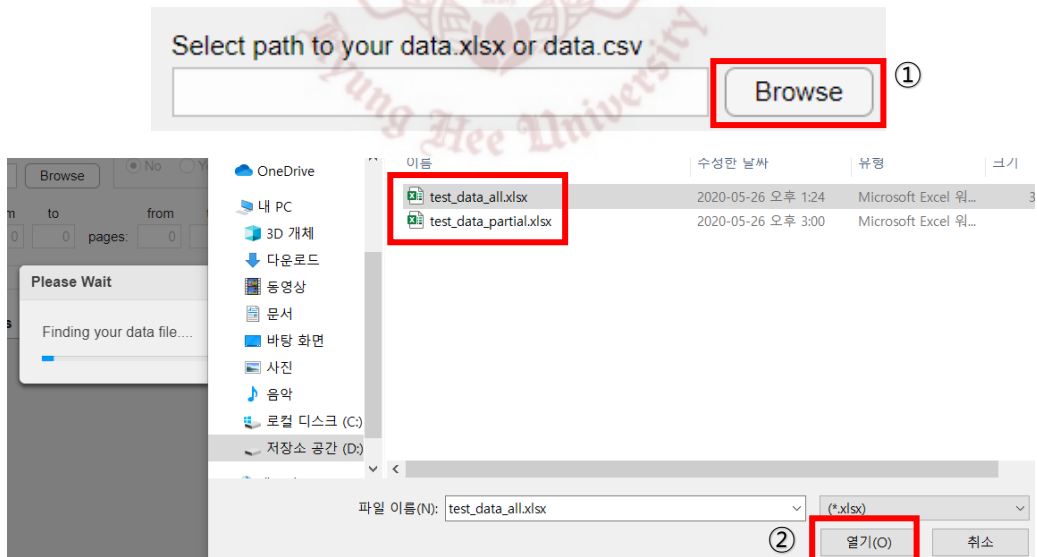


Figure 3.3. Browsing the dataset.

Once the data browsing process is complete, you can use the ‘Data Matrix Editor’ to resize the data array. Click ‘Yes’ in the editor to enable the editing as follows.

The figure consists of two screenshots of the 'Data Matrix Editor' interface. The top screenshot, labeled with a circled 1, shows the 'Data Matrix Editor' with the 'No' radio button selected. The bottom screenshot, labeled with a circled 2, shows the 'Data Matrix Editor' with the 'Yes' radio button selected. In this second screenshot, the 'rows', 'columns', and 'pages' fields are highlighted with a red box, showing values 1, 31, and 9 respectively. The 'rows' field has 'from' 1 and 'to' 100. The 'columns' field has 'from' 1 and 'to' 31. The 'pages' field has 'from' 1 and 'to' 9.

Figure 3.4. Usage of the ‘Data Matrix Editor’.

When ‘Data Matrix Editor’ is activated, the size of the selected data array is automatically entered. Here, you can enter the desired size in each field. As another feature, you can apply ‘scaling’ and ‘centering’ to a given dataset. Check the checkbox as shown in Figure 3.5. In this case, scaling and centering means satisfying the following conditions.

Let $X = (x_{ij}) \in \mathbb{R}^{p \times n}$ be a data matrix and $x_j = (x_{1j}, \dots, x_{pj})^T \in \mathbb{R}^p$ is a column vector or node j . Then x_j is centered and normalized (scaled) if $\sum_{i=1}^p x_{ij} = 0$ and $\|x_j\|^2 = x_j^T x_j = \sum_{i=1}^p x_{ij}^2 = 1$.

When all settings are complete, click the ‘Load’ button in the blue box, at the bottom right of the screen above. This saves the data array with the settings applied and calculates 3 distances and 1 minimum spanning tree required for data analysis. More specifically, It calculates ‘correlation-based distance matrix’, ‘single linkage matrix’, ‘Gromov-Hausdorff distance matrix’,

and minimum spanning tree. These can be viewed or saved directly in ‘Viewer’. See the related topic for ‘Viewer’.

The figure consists of two screenshots of the 'Data Matrix Editor' interface. The top screenshot shows the 'Load' button highlighted with a red box and a circled 2. The 'scaling and centering' checkbox is unchecked, and it is circled with a red box and a circled 1. The bottom screenshot shows the 'Initialize' button highlighted with a red box. The 'scaling and centering' checkbox is now checked. Both screenshots show the file path 'degree100_001_200021\data\test_data_all.xlsx' and the 'Data Matrix Editor' options set to 'No' for 'No' and 'Yes' for 'Yes'.

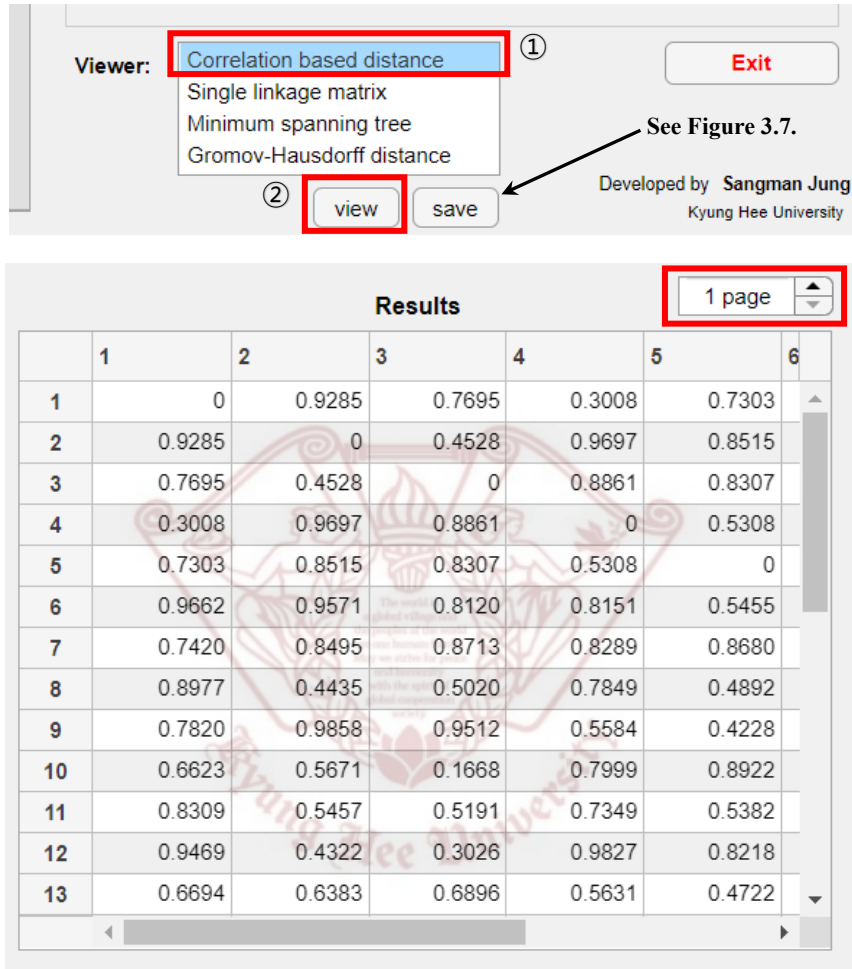
Figure 3.5. The process of loading the dataset. In this figure, the upper is the screen for the usage of loading the data with the checkbox ‘scaling and centering’. The lower is the screen for the button ‘Initialize’.

If you are not satisfied with the loaded data and its settings, click the 'Initialize' button marked with a red checkbox next to the 'Load' button to initialize all settings. This button becomes active when data loading is complete, and the rest are inactive. Also, the button initializes not only the loaded data but also all settings made when using the Toolbox in the future.

3.2.1.2 Usage of Viewer and Results

After loading the data, ‘Correlation-based distance’, ‘Single linkage matrix’, ‘Minimum spanning tree’, and ‘Gromov-Hausdorff distance’ are automatically calculated for the given data. These four values can be viewed through the ‘Viewer’ and saved as .xlsx format. If you click the

‘view’ button in Figure 3.6, you can see the calculated value as the table format, and if you click the ‘save’ button in Figure 3.7, the file is saved in the desired path as .xlsx format.



Viewer: **Correlation based distance** ①

- Single linkage matrix
- Minimum spanning tree
- Gromov-Hausdorff distance

② **view** **save** **Exit**

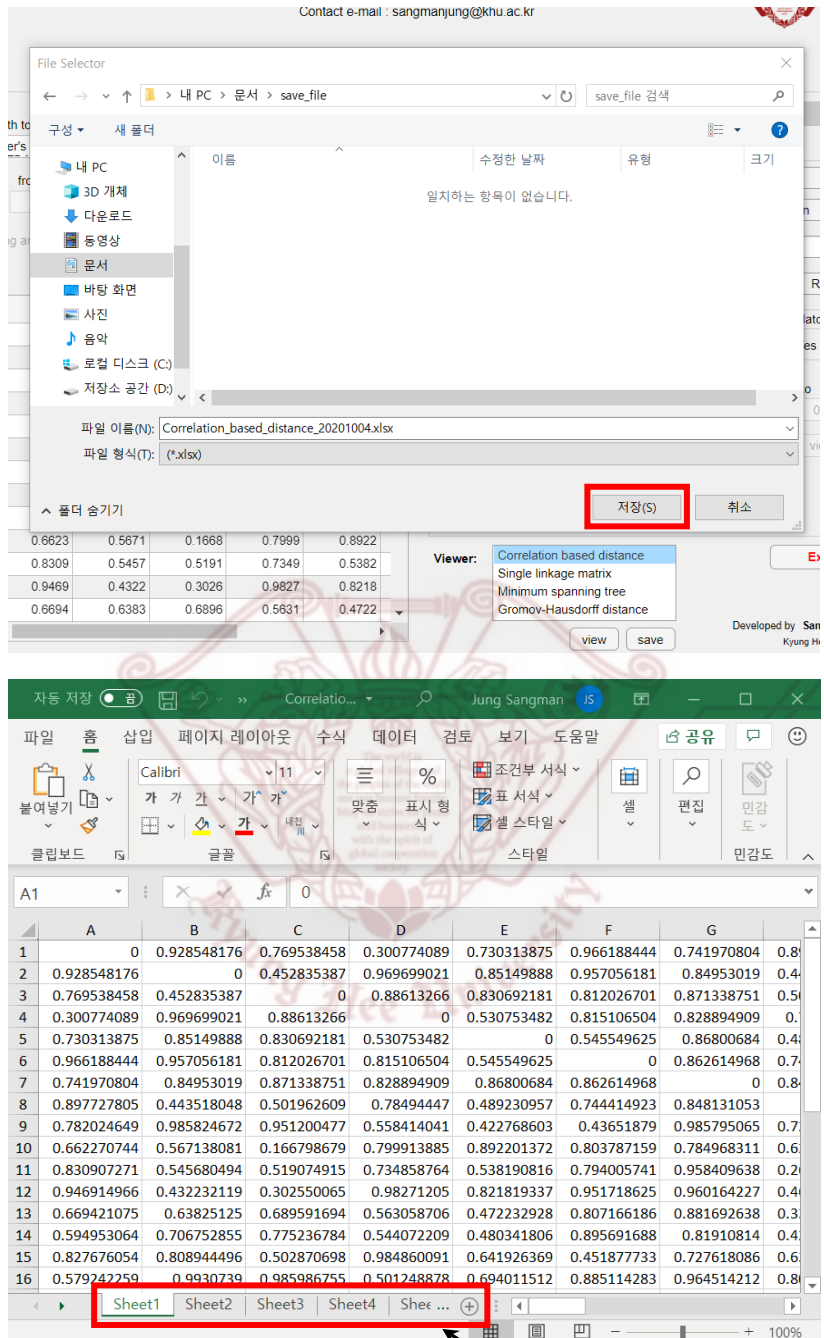
See Figure 3.7.

Developed by Sangman Jung
Kyung Hee University

Results 1 page

	1	2	3	4	5	6
1	0	0.9285	0.7695	0.3008	0.7303	
2	0.9285	0	0.4528	0.9697	0.8515	
3	0.7695	0.4528	0	0.8861	0.8307	
4	0.3008	0.9697	0.8861	0	0.5308	
5	0.7303	0.8515	0.8307	0.5308	0	
6	0.9662	0.9571	0.8120	0.8151	0.5455	
7	0.7420	0.8495	0.8713	0.8289	0.8680	
8	0.8977	0.4435	0.5020	0.7849	0.4892	
9	0.7820	0.9858	0.9512	0.5584	0.4228	
10	0.6623	0.5671	0.1668	0.7999	0.8922	
11	0.8309	0.5457	0.5191	0.7349	0.5382	
12	0.9469	0.4322	0.3026	0.9827	0.8218	
13	0.6694	0.6383	0.6896	0.5631	0.4722	

Figure 3.6. Usage of the button ‘view’ in the ‘Viewer’.



Save each page of the array as a sheet

Figure 3.7. Usage of the button 'save' in the 'Viewer'.

3.2.2 Data Analysis Tab

3.2.2.1 Hierarchical Clustering Toolbox Tab

‘HC Toolbox Tab’ is for hierarchical clustering (HC) which is one of the types of clustering methodology. In more detail, SLD and a heatmap, and a barcode curve plot of the 0th-Betti number are provided.

- *Heatmap*

After loading the data, it is ready to run the heatmap immediately. However, before executing this visualization tool, first, select the color of the heatmap and the matrix you want to see as the heatmap.

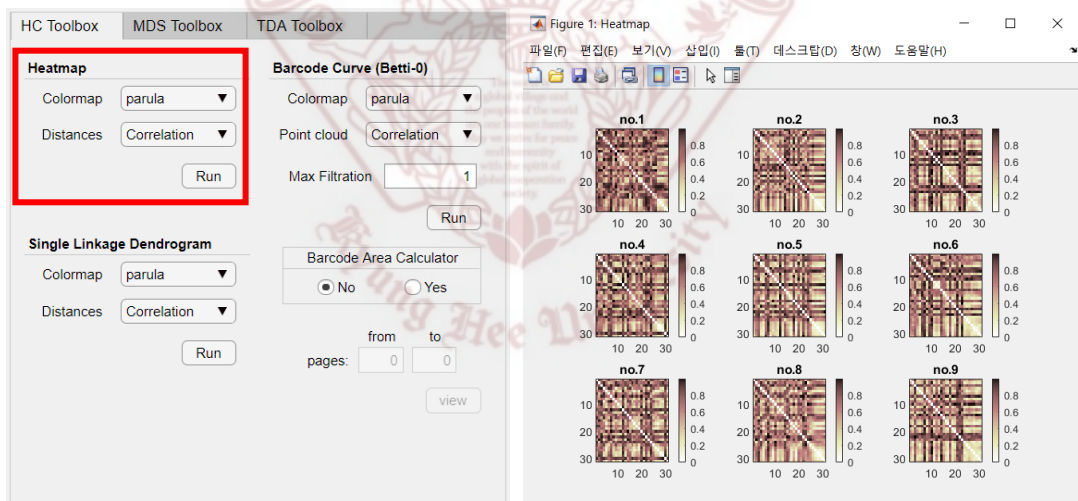


Figure 3.8. Usage of the heatmap tool. In this figure, the left is the location of the heatmap tool, and the right is the result when ‘Colormap’ is ‘pink’, and ‘Distances’ is a correlation-based distance matrix ($31 \times 31 \times 9$).

The heatmap tool automatically displays each page of the actual plot as 'no.i' when the data is an array. For 'Distances', correlation-based distance, single linkage distance, and GH distance can be selected by default. If BN distance is calculated, it is added to the 'Distances' item. 'Colormap' has 18 colormaps provided by MATLAB by default. The color table presented in MATLAB Help is attached below.

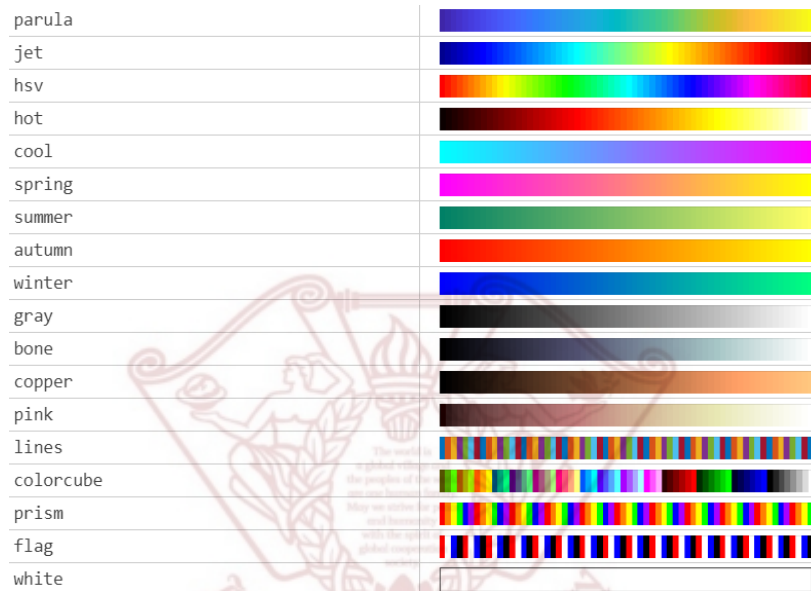


Figure 3.9. The colormaps in MATLAB.

- *Single Linkage Dendrogram*

The method of using the SLD tool is the same as the heatmap tool. The figure obtained as actual execution is as follows.

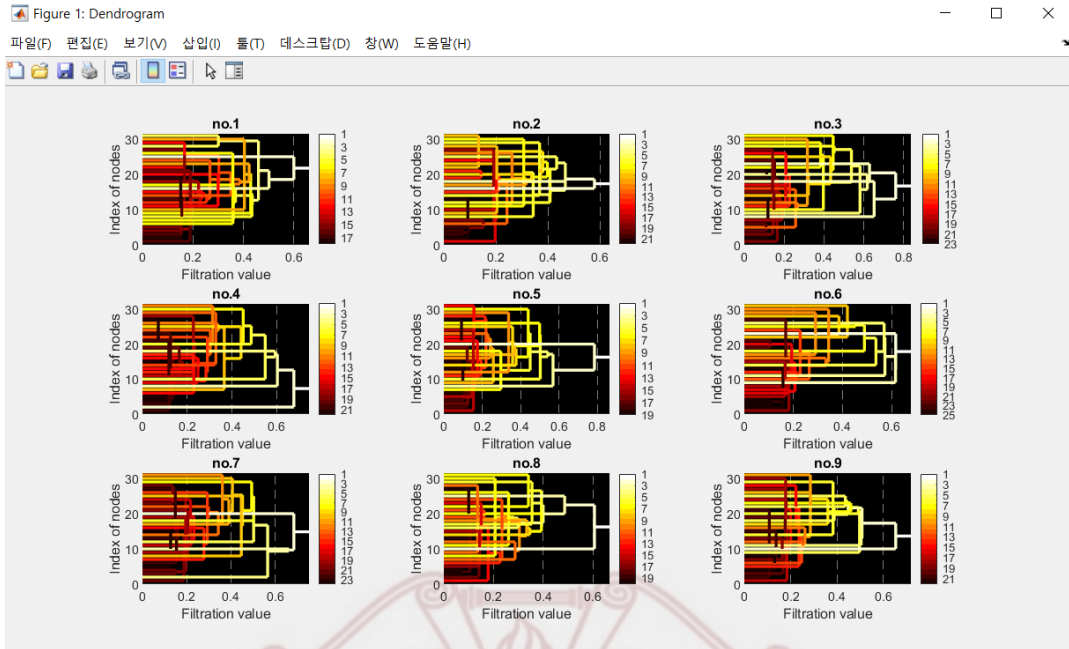


Figure 3.10. The result of the single linkage dendrogram when ‘Colormap’ is ‘hot’, and ‘Distances’ is a correlation-based distance matrix ($31 \times 31 \times 9$).

• Barcode Curve

The barcode curve shows the barcode as a curve using the fact that the single linkage dendrogram is equivalent to the barcode for the 0th Betti number. More information about this can be found in [18, 19]. When the barcode is represented as a curve, it is easier to compare among the barcode, and there is an advantage that it can be quantified by a simple method of obtaining the area. In Figure 3.11., the popup menu ‘Point cloud’ is the same as the popup menu ‘Distances’ like other tools that have ‘Distances’. The edit field ‘Max Filtration’ means the filtration value of the barcode, and only real values that are positive numbers greater than or equal to 0 can be entered. The ‘Barcode Area Calculator’ at the bottom of the panel operates independently of the barcode curve tool and calculates the barcode area for each page of the array. If you press the ‘Yes’ button on the calculator, the edit field for the page below it becomes active. See Figure 3.12.

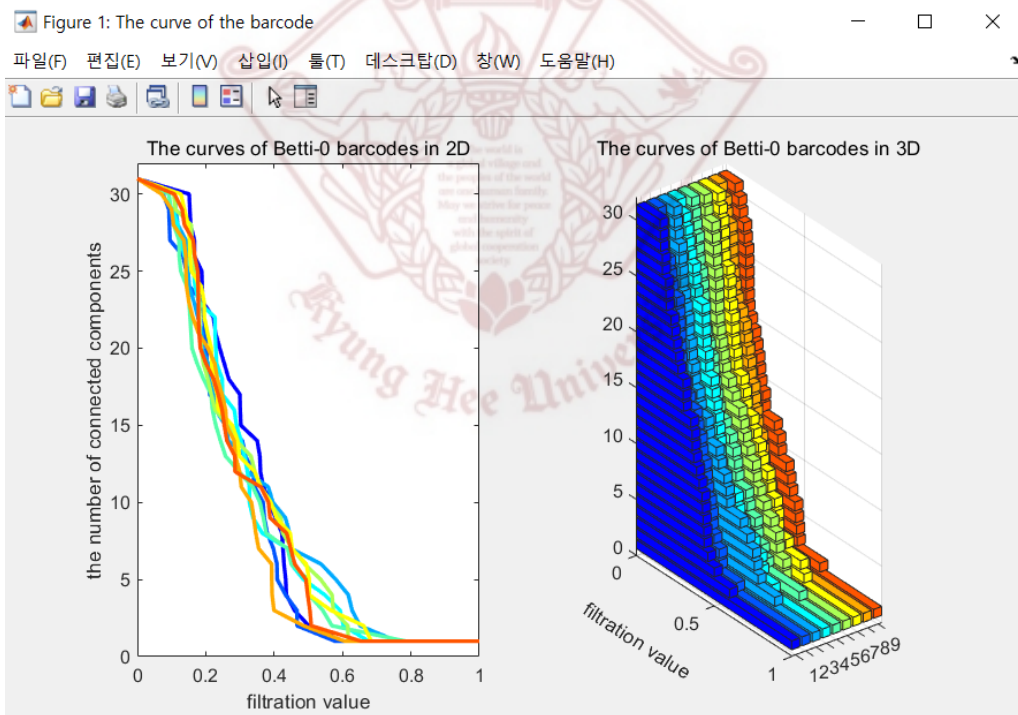
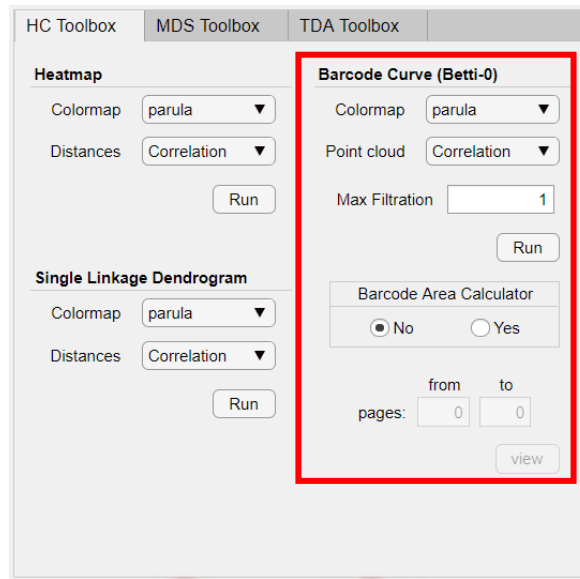


Figure 3.11. The ‘Barcode Curve (Betti-0)’ panel (top), and its result (bottom), when ‘Colormap’ is ‘jet’, ‘Point cloud’ is correlation-based distance matrix, ‘Max Filtration’ is 1.

Barcode Area Calculator

☐ No ☒ Yes ^①

② pages: from 1 to 9

Results

Area for each barcode =

9.9265
8.8016
9.9910
9.6193
8.9124
9.7992
9.9360
8.4916
9.3976

Mean of barcode areas =

9.4306

Figure 3.12. Usage of the ‘Barcode Area Calculator’. The left is the figure that the calculator is activated, and the right is the results of the calculator as clicking the ‘view’ button.

3.2.2.2 Multidimensional Scaling Toolbox Tab

‘MDS Toolbox Tab’ is for multidimensional scaling (MDS) which is one of the types of dimensionality reduction methods. This toolbox provides the Shepard plot (for MDS), scree plot (for k-means clustering), MDS, and k-means clustering. The Shepard plot is used to improve the goodness of fit of MDS, and the scree plot is used to find the optimal ‘k’ value in k-means clustering.

- *Shepard plot*

The Shepard plot is to determine how well the model fit of MDS is, and it is a method to check the difference between input and output by placing the horizontal axis as the input distance and the vertical axis is the output distance. The 'stress', which has the meaning of error in MDS, also plays an important role in determining the goodness of fit, but even if the stress value is small, the output MDS plot cannot be considered a well-expressed result. This program supports the Shepard plot for 6 loss function models of MDS. The execution screen of this plot is as shown in Figure 3.13.

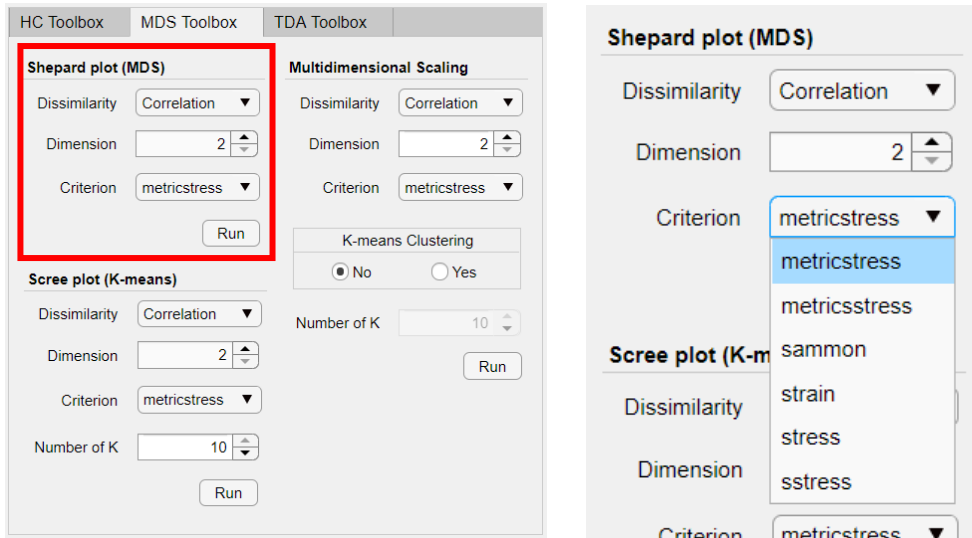


Figure 3.13. The Shepard plot tool of MDS Toolbox Tab.

In the Shepard plot, the 'Dimension' means the dimension to be reduced in MDS. The dimension value is usually 2 or 3 and sometimes results that more than 4 dimensions are checked, so the default value is from 2 to 10. Figure 14 shows the actual execution results.

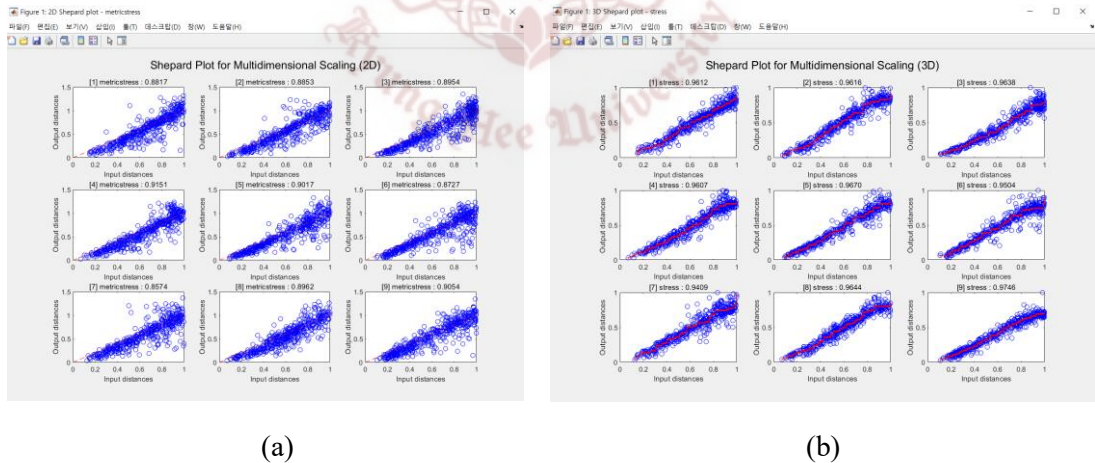


Figure 3.14. Two Shepard plot results for each page of the correlation-based distance array. (a) is the result of the Shepard plot for 2D, using the loss function 'metricstress', and (b) is the result of the Shepard plot for 3D, using the loss function 'stress'.

In Figure 3.14, (a) is a loss function that is metric scaling, and (b) is a loss function that is non-metric scaling, and a red fitting curve is changed according to each case. In each subplot of the Shepard plots, the title has the following meaning in order from left to right: $[i]$ is the i th page of the input array, the loss function used, and the Pearson correlation coefficient.

- *Scree plot*

The scree plot is a method commonly used in exploratory data analysis to determine the key factor of a specific parameter. This method can be used not only in k-means clustering but also in MDS (eg, when determining dimensions). However, this program was limited to k-means clustering. The usage and each selection item are the same as the Shepard plot, but the spinner 'Number of K' that determines the k value of k-means clustering is added.

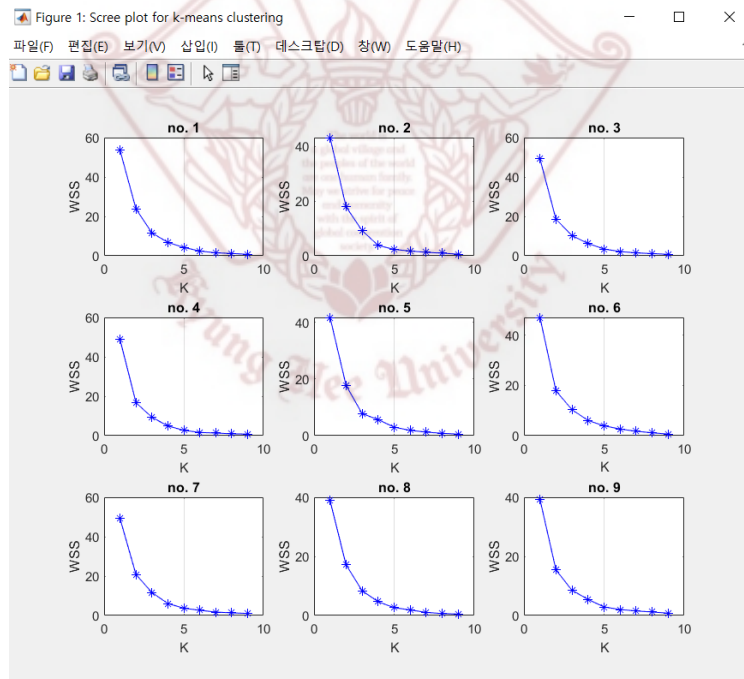
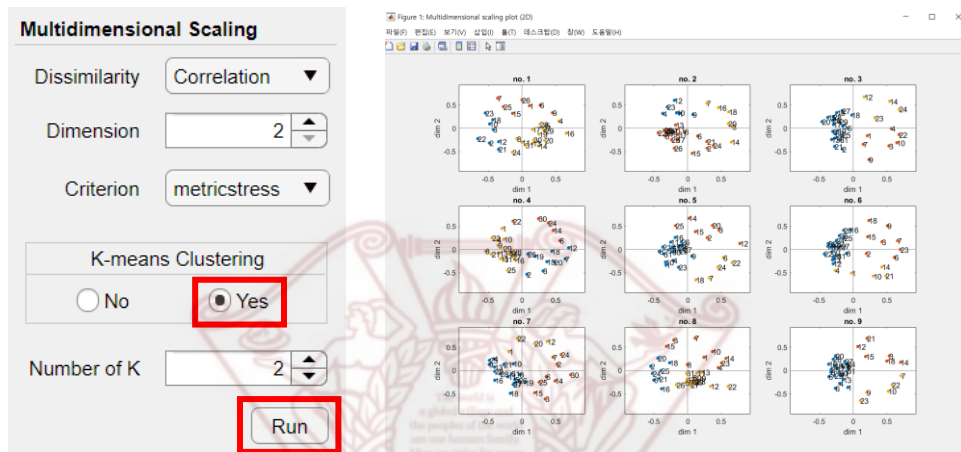


Figure 3.15. The scree plot for 3D MDS for the criterion ‘metricstress’. In this case, $k = 9$.

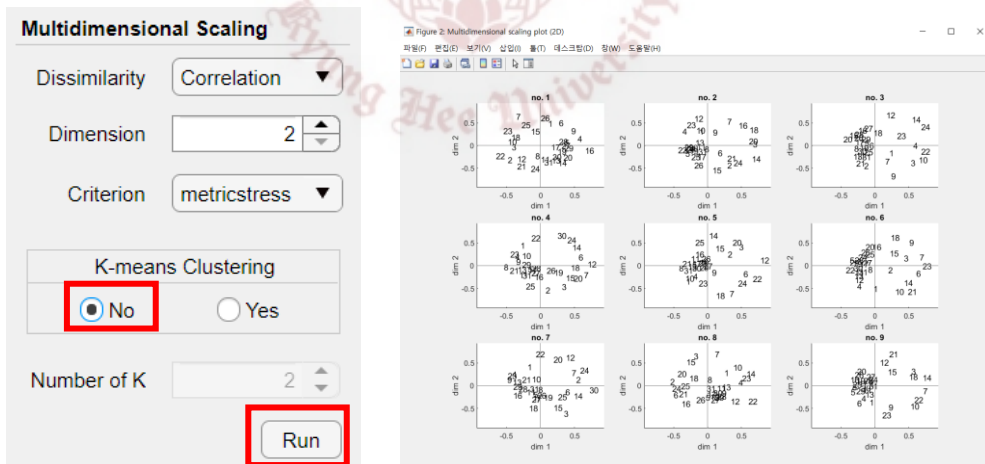
In Figure 3.15, the 3D MDS result for each page (subject) of the input array indicates that at least the number of clusters should be $k = 3$.

- *Multidimensional Scaling*

MDS is a technique that expresses the dissimilarity between variables in a lower dimension for a given dissimilarity matrix. It is recommended to execute the MDS after determining the loss function and dimension of the MDS using the Shepard plot described above. Also, if you want to apply k-means clustering, it is recommended to check the scree plot first.



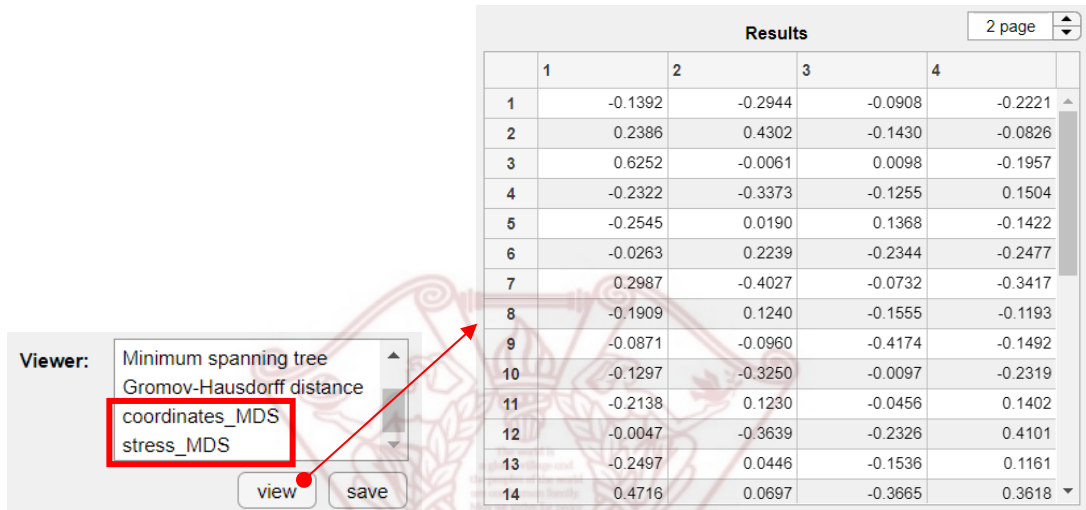
(a) The 2D MDS plot if the option 'K-means Clustering' is selected 'Yes'



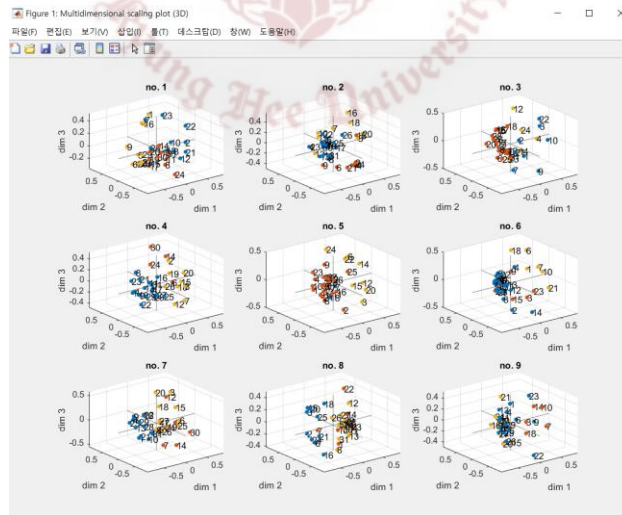
(b) The 2D MDS plot if the option 'K-means Clustering' is selected 'No'

Figure 3.16. The results of the MDS tool.

As shown in Figure 3.16 (a) and (b), colored points appear or disappear in the plot depending on whether k-means clustering is used. Also, after executing the MDS, the MDS coordinates and stress, which are errors, are stored in the 'Viewer', and the values can be viewed or saved with the 'view' button and the 'save' button. If the dimension exceeds three dimensions, it is not plotted, and only coordinate values and stresses are stored. See Figure 3.17.



(a) The results of 4D MDS stored in 'Viewer'



(b) The 3D MDS plot adopting the k-means clustering

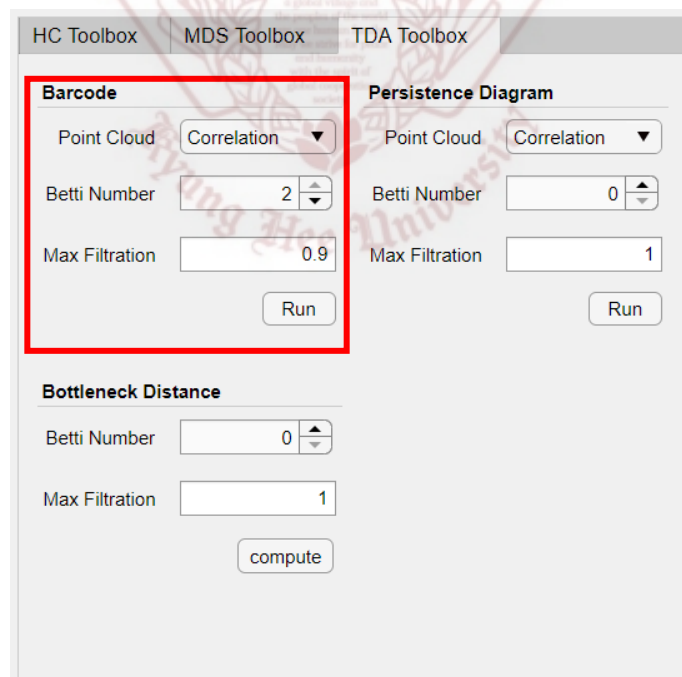
Figure 3.17. The MDS results at 3D or more.

3.2.2.3 Topological Data Analysis Toolbox Tab

We introduce ‘TDA Toolbox’ which has the main techniques of TDA. Such a toolbox contains the barcode and the PD. We also add the BN distance and describe how to use the BN distance to quantitatively compare the difference between the two PDs. In this program, TDA Toolbox was developed using 'Javaplex', an open-source excellent for calculating persistent homology. Therefore, it is recommended that anyone who wants to access the source code of this program learn how to use the Javaplex [11]. All methods in this toolbox use the Vietoris-Rips filtration.

- *Barcode*

To calculate the barcode, the 'point cloud' which is the data to be analyzed, the dimension of the Betti number to be viewed, and the maximum filtration value must be determined. The filtration interval starts from 0.



The screenshot displays the 'TDA Toolbox' tab in a software interface. It features three main panels: 'Barcode', 'Persistence Diagram', and 'Bottleneck Distance'. The 'Barcode' panel is highlighted with a red rectangular box. Within this panel, there are three input fields: 'Point Cloud' with a dropdown menu set to 'Correlation', 'Betti Number' with a numeric input set to '2', and 'Max Filtration' with a numeric input set to '0.9'. A 'Run' button is located at the bottom of the 'Barcode' panel. The 'Persistence Diagram' panel to the right has similar controls, with 'Betti Number' set to '0' and 'Max Filtration' set to '1'. The 'Bottleneck Distance' panel at the bottom has 'Betti Number' set to '0' and 'Max Filtration' set to '1', with a 'compute' button.

Figure 3.18. The barcode panel in TDA Toolbox Tab.

In the barcode panel of Figure 3.18, Let the point cloud is set to the correlation-based distance, the maximum dimension of the Betti number is set to 2, and the filtration interval is set to $[0, 0.9]$. The implementation result is as shown in Figure 3.19.

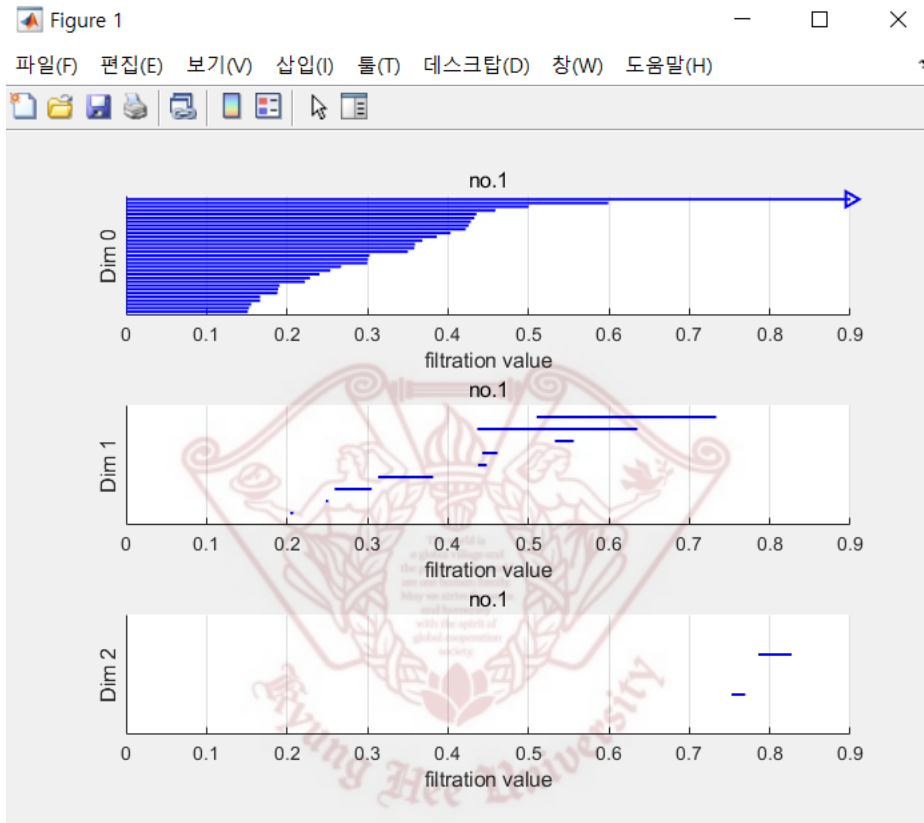


Figure 3.19. The β_k barcode plot for $k = 0$ (the upper), $k = 1$ (the middle), $k = 2$ (the bottom).

- *Persistence Diagram*

The PD is a 2-dimensional scatter plot that showing information on ‘birth’ and ‘death’ of topological features for the point cloud. The usage of this diagram is the same as that of the barcode. Let the point cloud is set to the correlation-based distance, the maximum dimension of

the Betti number is set to 2, and the filtration interval is set to $[0, 1.1]$. The implementation result is as follows.

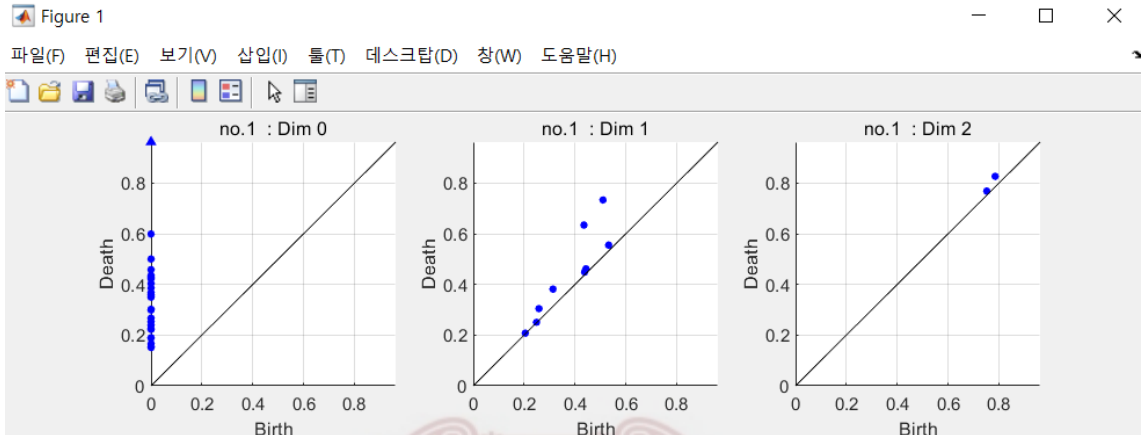


Figure 3.20. The persistence diagram for β_k , where $k = 0$ (the left), $k = 1$ (the center), $k = 2$ (the right).

- *Bottleneck distance*

The BN distance is the measure to calculate the difference between two PDs. The usage of this measure is the same as that of the barcode or PD except for the selection of ‘Point Cloud’. This tool supports only the correlation-based distance matrix. After the computation was completed, the result of the BN distance is added to the list of ‘Viewer’. See Figure 3.20.

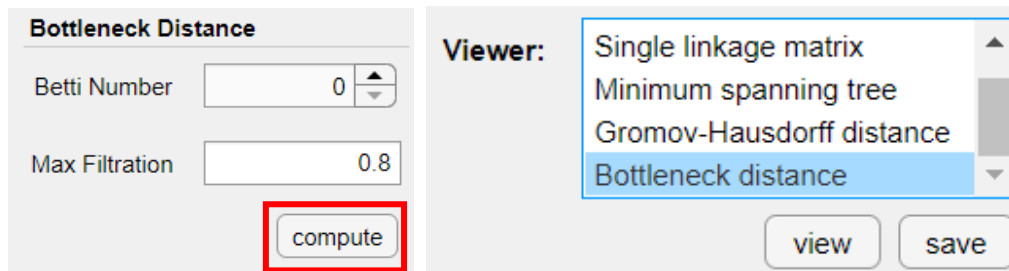
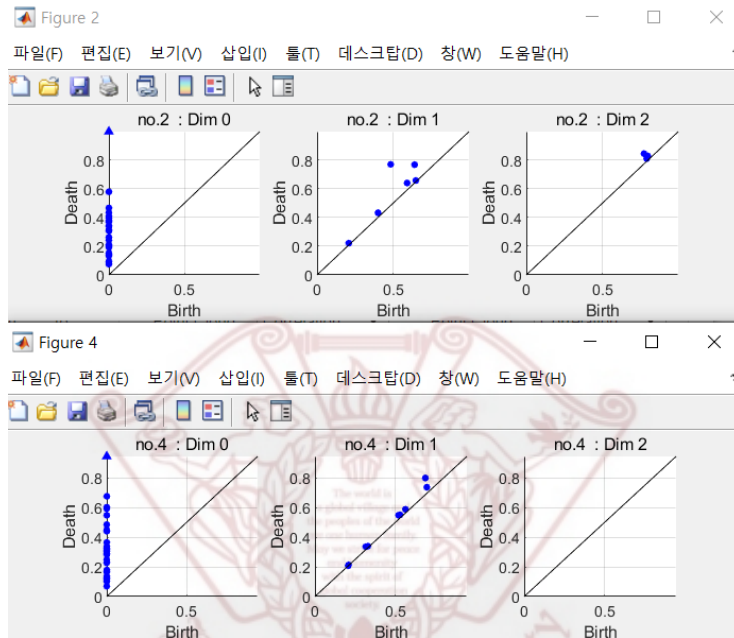
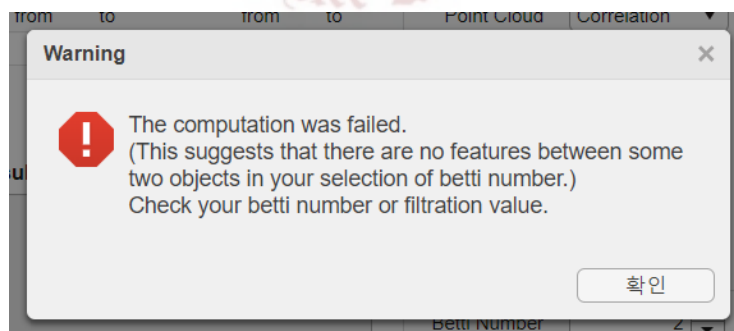


Figure 3.21. The computation result of the bottleneck distance.

If one of two or more PDs has no topological feature at specific β_k , the computation canceled. For example, if the data array has 9 pages and we want to see the result of β_2 , but one of 9 pages has no topological feature at β_2 , then the warning message occurs on the screen as shown in Figure 3.21.



(a) Two persistence diagrams 'no.2' and 'no.4'.



(b) The warning message.

Figure 3.22. The warning message for computing the bottleneck distance.

Chapter 4. Application of Topological Data Analysis

In this Chapter, we applied the TDA with some conventional data analysis methods such as SLD, MDS, k-means clustering. Also, we computed two different distances which are the bottleneck distance and GH distance.

4.1 Biomechanical dataset

We used the dataset obtained by measuring the joint angles of 10 body parts of 3 movements for 9 golfers (subjects) in the swing motion from the ready position to the motion of hitting the golf ball. We normalized the swing motion to 100 seconds as the observations. In addition, X-factor (Xf) which is the value of the difference between the rotation angles for the upper and lower bodies while the golfer takes a backswing is considered. Thus, the size of this dataset as the three-dimensional array is seconds \times joint angles + X-factor \times subjects = $100 \times 30 + 1 \times 9$. The time interval is divided into three parts: Take back (1~47 seconds), ball impact (48~65 seconds), follow up (66~100 seconds). Also, all the joint angles are divided into three groups: Flexion-Extension (FE), Abduction-Adduction (AD), Internal-External rotation (IE). Each group has 10 body parts, and 8 out of 10 body parts are divided into two parts: the left part (1~4) and the right part (5~8), and each part has four parts as a shoulder, elbow, hip, knee. The rest are an upper trunk and a lower trunk (9,10). We used the correlation-based distance as a metric for constructing point clouds. This distance becomes a metric under certain conditions as follows [40].

Proposition 4.1. Let X and Y are random variables that are centered and normalized. Then the correlation-based distance defined by $d_c(x, y) = (1 - |\rho_{xy}|)^{1/2}$ where ρ_{xy} is the Pearson correlation coefficient between $x \in X$ and $y \in Y$ is a metric.

Thus, for $i = 1, \dots, 9$, let $X_i \in \mathbb{R}^{100 \times 31}$ be a data matrix, then we obtain 9 (dis)similarity matrix $d_c^{(i)}$ as the point clouds by Proposition 4.1 above, and the size of $d_c^{(i)}$ becomes 31×31 .

4.2 Results

The result of the heatmap of the point clouds for 9 subjects is as shown in Figure 4.1. Each movement has 10 variables as the left or right shoulder (LS / RS), elbow (LE / RE), hip (LH / RH), knee (LK / RK), and trunk (LT / RT). Each color of the cell corresponds to each value of $d_C^{(i)}(x_{row}, x_{column})$ which has the value from 0 (blue) to 1 (red). The value 0.5 has a white color in the heatmap.

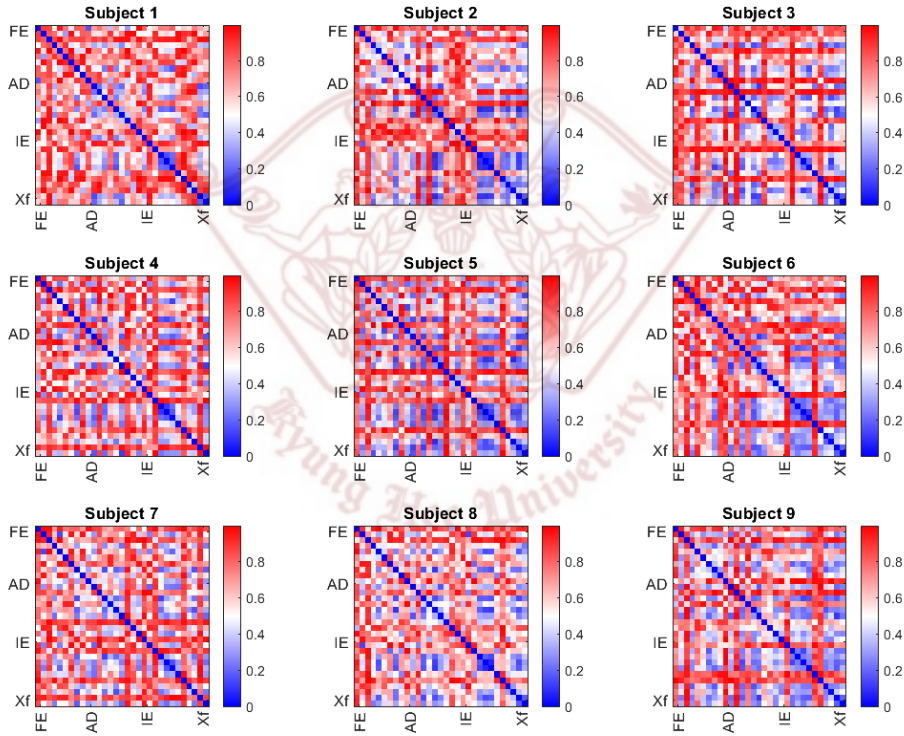


Figure 4.1. The heatmap result of the correlation-based distance matrix d_C for 9 subjects.

We represented the barcode for β_0 as a curve and computed the area and slope of the curve of the barcode. Figure 4.2 shows the curve representation of the barcode for β_0 of the point cloud

for each subject. Since we used the correlation-based distance, the minimum filtration value is 0, and the maximum filtration value is 1. The obtained areas of the barcodes for β_0 are 10.11, 8.99, 10.17, 9.8, 9.1, 9.98, 10.12, 8.68 and 9.58 for subject 1 through subject 9. The mean and standard deviation for these areas are obtained as 9.61 and 0.56. The decreasing slopes are 71.8, 65.2, 44.4, 53.5, 47.6, 52.5, 52.4, 68.3, and 57.9. The filtration values when all connected components are merged are 0.6, 0.578, 0.759, 0.676, 0.784, 0.616, 0.684, 0.606, and 0.654 for subject 1 through subject 9. All this information of the barcode for β_0 is summarized in Table 4.1. For the case of β_1 , we computed the BN distance only. See Table 4.3.

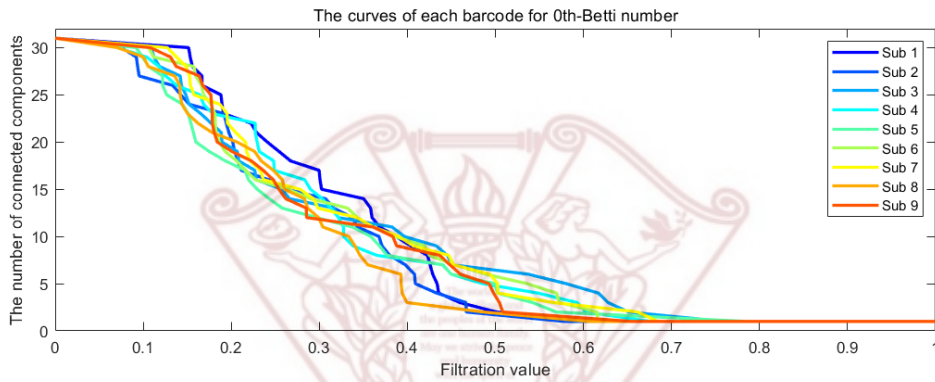


Figure 4.2. The curve representation of the barcode (β_0) for each subject. The vertical and horizontal axes represent the number of connected components and filtration values.

Table 4.1. The area and slope of the curve of the barcode (β_0) for 9 subjects. The row ‘End_F’ is the end of the filtration which means when all connected components are merged.

	Sub1	Sub2	Sub3	Sub4	Sub5	Sub6	Sub7	Sub8	Sub9	Mean	STD
Area	10.11	8.99	10.17	9.8	9.1	9.98	10.12	8.68	9.58	9.61	0.56
Slope	71.8	65.2	44.4	53.5	47.6	52.5	52.4	68.3	57.9	57.1	9.4
End_F	0.6	0.578	0.759	0.676	0.784	0.616	0.684	0.606	0.654	0.662	0.071

The result of the heatmap of the single linkage matrix in which the filtration for the 0th-Betti number is converted into a matrix using the single linkage method is shown in Figure 4.3.

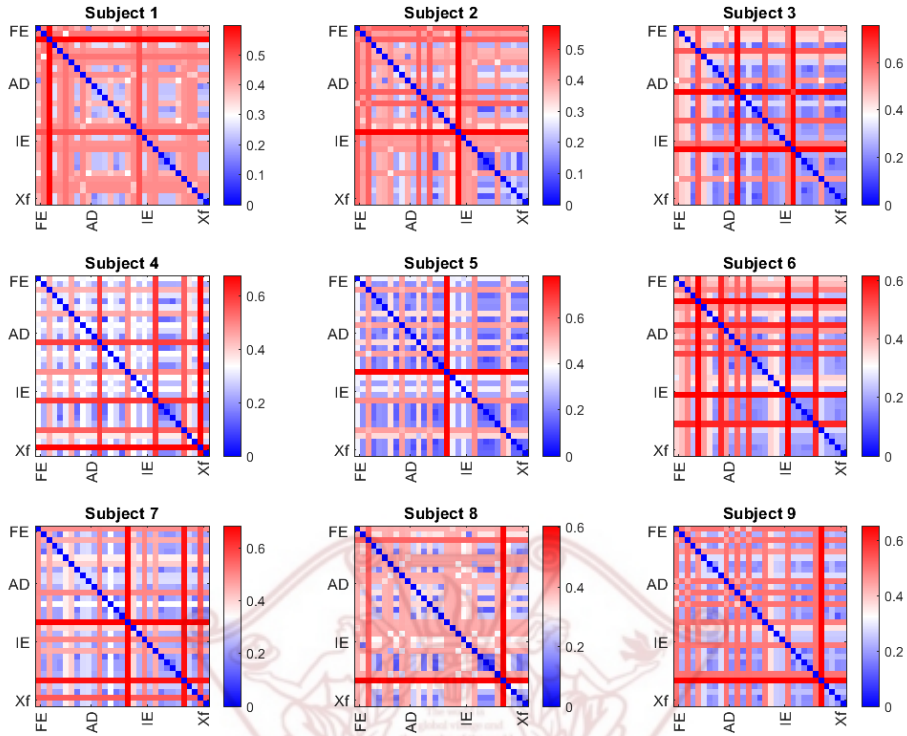


Figure 4.3. The heatmap result of the single linkage matrix d_S for 9 subjects.

For the single linkage matrix, the value of each cell means the filtration value. The dendrogram for d_S is presented in Figure 4.4. The vertical and horizontal axes are the node index or variable and filtration value. The color bar of each dendrogram shows the distance to the giant component. The distance to the giant component of the giant component is 1. Whenever, the connected component is divided into smaller components, the distance increase one by one. The maximum distances to the giant component are computed as 17, 21, 22, 21, 18, 24, 23, 19, 21.

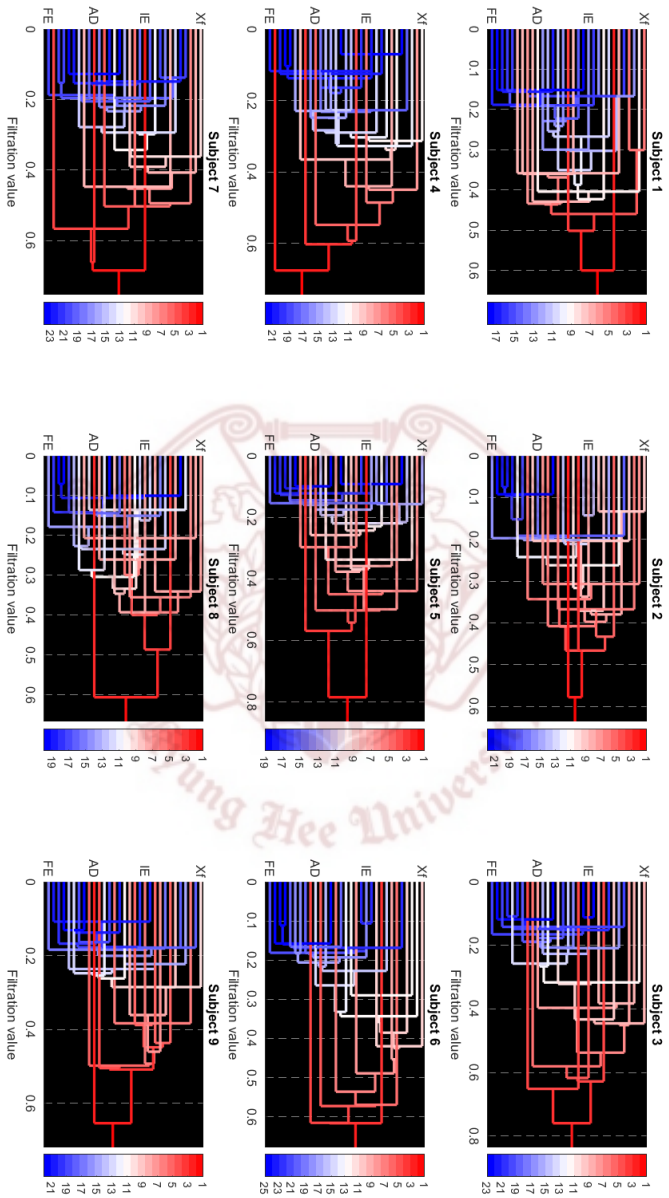


Figure 4.4. The single linkage dendrogram for 9 subjects. The vertical axis represents 31 connected components as FE (10), AD (10), IE (10), Xf (X-factor). The horizontal axis represents filtration value.

We computed the pairwise BN distance for each PD for β_0 and β_1 . For the case of β_0 , we also computed the GH distance using the single linkage matrix d_S . The results of these are as shown in Table 4.2, Table 4.3, Table 4.4, respectively. We set the maximum filtration value is 1.

Table 4.2. The Bottleneck distance matrix (β_0) for 9 subjects.

BN (β_0)	Sub1	Sub2	Sub3	Sub4	Sub5	Sub6	Sub7	Sub8	Sub9
Sub1	0								
Sub2	0.088	0							
Sub3	0.182	0.185	0						
Sub4	0.133	0.136	0.095	0					
Sub5	0.185	0.206	0.115	0.108	0				
Sub6	0.134	0.149	0.143	0.081	0.168	0			
Sub7	0.16	0.195	0.115	0.082	0.1	0.068	0		
Sub8	0.076	0.067	0.228	0.193	0.178	0.175	0.175	0	
Sub9	0.089	0.083	0.143	0.094	0.13	0.107	0.153	0.105	0

Table 4.3. The Bottleneck distance matrix (β_1) for 9 subjects.

BN (β_1)	Sub1	Sub2	Sub3	Sub4	Sub5	Sub6	Sub7	Sub8	Sub9
Sub1	0								
Sub2	0.0995	0							
Sub3	0.0995	0.7800	0						
Sub4	0.1115	0.1430	0.1085	0					
Sub5	0.0995	0.0890	0.0220	0.1115	0				
Sub6	0.0995	0.0420	0.0890	0.1485	0.1000	0			
Sub7	0.0995	0.0940	0.0660	0.0940	0.0720	0.1050	0		
Sub8	0.0995	0.0635	0.0510	0.1070	0.0680	0.0575	0.0720	0	
Sub9	0.0980	0.0635	0.0170	0.1090	0.0320	0.0730	0.0720	0.0360	0

Table 4.4. The Gromov-Hausdorff distance matrix for 9 subjects.

GH	Sub1	Sub2	Sub3	Sub4	Sub5	Sub6	Sub7	Sub8	Sub9
Sub1	0								
Sub2	0.3261	0							
Sub3	0.5371	0.4161	0						
Sub4	0.5246	0.5839	0.5576	0					
Sub5	0.4851	0.4475	0.4624	0.5542	0				
Sub6	0.2729	0.4117	0.5776	0.5186	0.5919	0			
Sub7	0.4145	0.4739	0.4901	0.2755	0.4441	0.4914	0		
Sub8	0.3238	0.4117	0.5529	0.5346	0.3922	0.4798	0.4245	0	
Sub9	0.3028	0.4598	0.6285	0.5680	0.4381	0.3896	0.4580	0.3622	0

For the BN distances for each dimension and the GH distance, as inputs, we visualized the results of these three distance matrices using MDS. We first checked the Shepard plot for MDS to reduce the error between the input distance and output distance. In Figure 4.5, we represent the Shepard plot of the GH distance for the 3-dimensional MDS result. The vertical and horizontal axes are the output of MDS and the input of MDS. In this figure, ‘D’, ‘C’, ‘P’ means ‘dimension’, ‘criterion’, and the ‘Pearson correlation coefficient’. We set the parameters of MDS as ‘D’ = 3, ‘C’ = ‘strain’. For the cases of the BN distances for β_0 and β_1 , we set ‘D’ = 3, ‘C’ = ‘metricstress’ for both cases, and their correlation coefficients are 0.9746 and 0.9802. We also determined $k_{BN0} = 3$, $k_{BN1} = 3$, $k_{GH} = 3$ for the bottleneck distance for β_0 and for β_1 , and GH distance, for k-means clustering. See Figure 4.6.

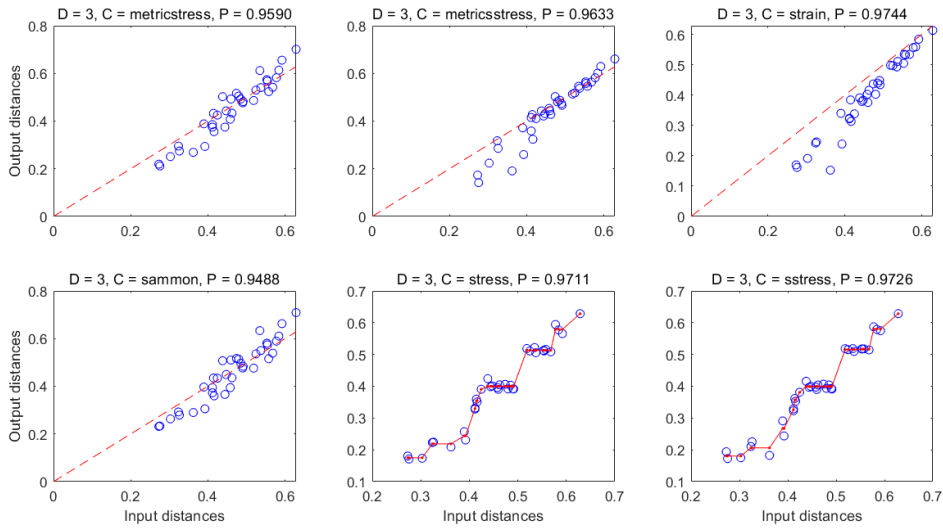


Figure 4.5. The Shepard plot of the Gromov-Hausdorff distance for the 3D MDS. ‘D’, ‘C’, ‘P’ means ‘dimension’, ‘criterion’, and the ‘Pearson correlation coefficient’.

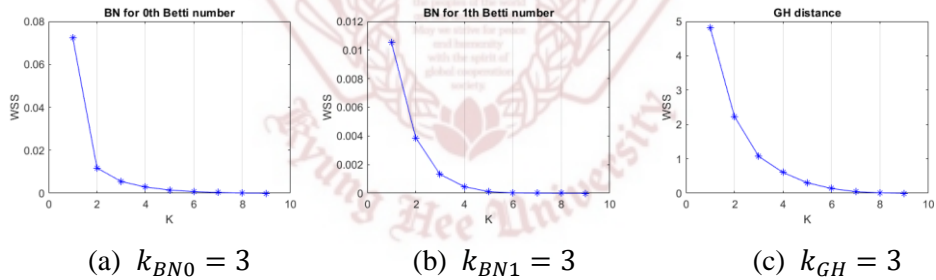
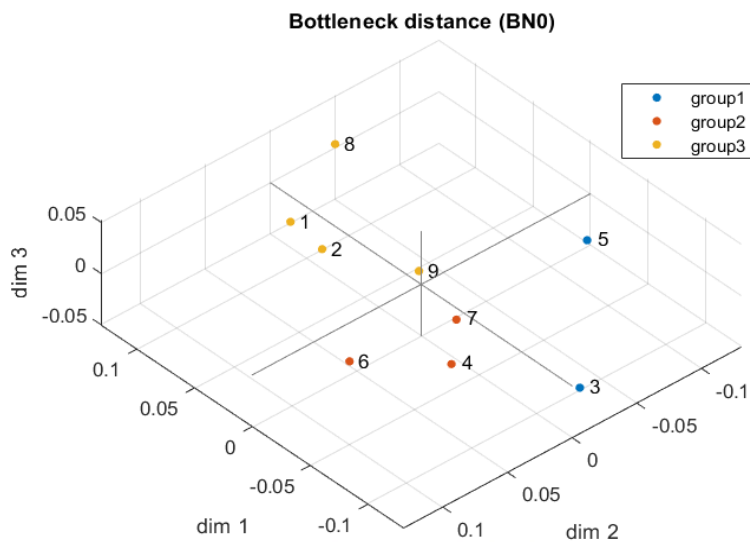
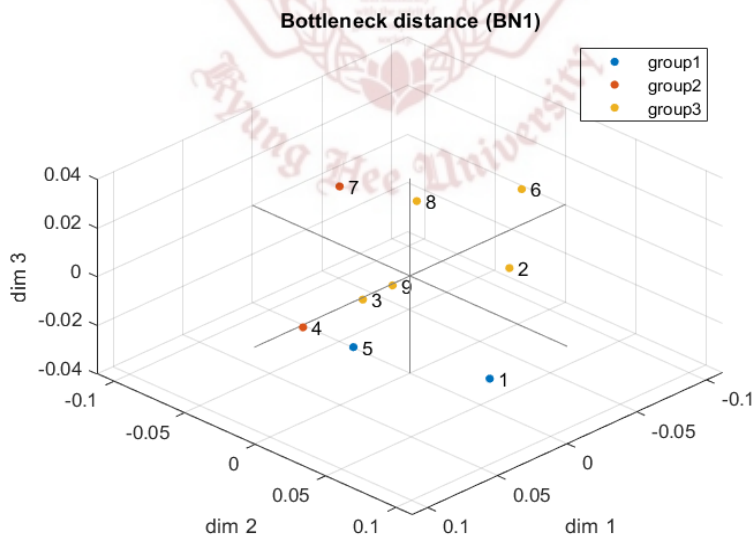


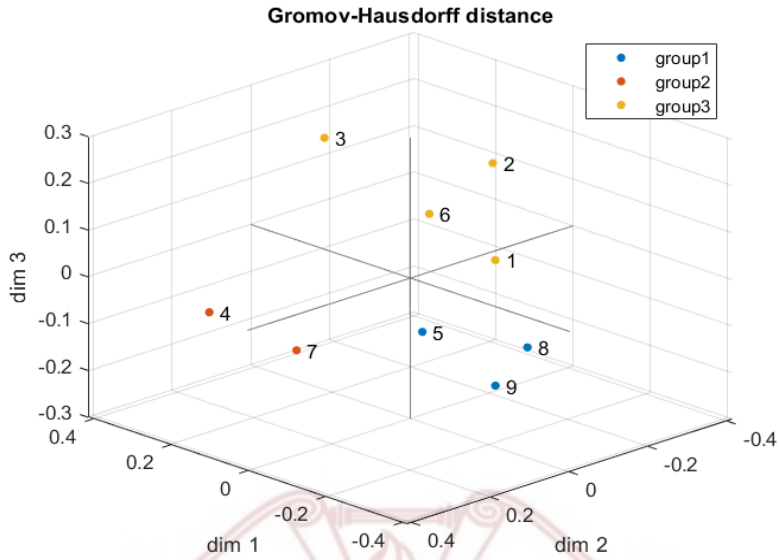
Figure 4.6. The scree plot of three distances for the 3D MDS. (a) is the bottleneck distance for β_0 , (b) is the bottleneck distance for β_1 , and (c) is the Gromov-Hausdorff distance. We set the optimal k as $k_{BN0} = 3$, $k_{BN1} = 3$, $k_{GH} = 3$, respectively



(a) The result of 3D MDS of the bottleneck distance for β_0 . The goodness of fit ('metricstress') is 0.0720.



(b) The result of 3D MDS of the bottleneck distance for β_1 . The goodness of fit ('metricstress') is 0.0757.



(c) The result of 3D MDS of the Gromov-Hausdorff distance. The goodness of fit ('strain') is 0.0963.

Figure 4.7. The k-means clustering results of 3D MDS for the bottleneck distance and the Gromov-Hausdorff distance.

Figure 4.7 shows the k-means clustering results of the 3-dimensional MDS. Each color of the points in the figure represents each group, and each number is the index of each subject. In (a), 5, 3 are classified as group 1 (blue), 4, 6, 7 are classified as group 2 (red), and 1, 2, 8, 9 are classified as group 3 (yellow). In the same way, 1, 5 are classified as group 1, 4, 7 are classified as group 2, and 2, 3, 6, 8, 9 are classified as group 3 in (b). In (c), 5, 8, 9 are group 4, 7 are group 2, and 1, 2, 3, 6 are group 3. In the case of (a), the points are divided into two groups such that group Y (3, 4, 5, 6, 7) and group B (1, 2, 8, 9) if we set $k_{BN0} = 2$.

4.3 Discussion

The result of the barcode for β_0 shows that subject 3 is the least connected and subject 8 is the fastest. In the case of subject 1, the overall connectivity of all nodes is degraded, but unlike subject 3, the topological feature disappears evenly. Also, we found the result that the difference between subjects is increasing from the filtration value 0.4 to 0.65. This implies that only 1 through 10 connected components are dominant at the difference between subjects.

For the SLD, the maximum distance to the giant connected component which means the complexity of the connectivity is subject 6, and the minimum of that is subject 1. It shows that subject 1 has a simple network structure rather than other subjects. In addition, most subjects have the nodes {8,9,10} (FE), {14,18,20} (AD), {25,26,29,30} (IE), {31} (Xf) that is the distance to the giant component from 1 to 11. This reveals that the connection speed of the left body parts for three groups (FE, AD, IE) is faster than the case of the right body parts. The result of this dendrogram, along with the result of the curve of the barcode, indirectly proves that the right body parts cause the differences between subjects.

As a result of cluster analysis of MDS for the three distances, no largely significant similarity or dissimilarity was found between subjects. That is, there is no distinct tendency in each group. However, we found that k-means clustering result of the BN distance for β_0 , β_1 and the GH distance shows that subject 4 and 7 are grouped and that subject 8 and 9 are grouped in common. Also, subject 1 and 2 are grouped in the results of BN distance for β_0 and GH distance. Besides, subject 1 and 2 in the MDS for the BN distance for β_1 are similar positions for each other.

Consequently, the approach of the concept of graph filtration in [19] is useful to detect the relationship between the variables or the nodes, and finding the difference between subjects is possible by visualizing the (dis)similarity between subjects individually using MDS.

Chapter 5. Conclusion

In this study, we applied TDA which is the new framework of data analysis based on the computational topology to the actual dataset. This methodology can find a hidden topological feature in the data and use it to obtain the quantified (dis)similarity from the dataset. However, one of the limitations of TDA is the computational aspect. Although a lot of software is available, users who unfamiliar with topology or programming still difficult to employ this methodology. Hence, we developed the GUI program for TDA in the MATLAB environment for these users. This program focused on the convenience to use and the accessibility and supports not only the methods in TDA but also the clustering methods and multivariate analysis such as hierarchical clustering, k-means clustering, MDS.

There is also a limitation of the analysis aspect. In Chapter 4, using TDA, we analyzed joint kinematics in biomechanics for 9 subjects with 31 variables, but there was no largely significant similarity between subjects as a result. This is because TDA gives us only the global topological information of the data. However, we also found the features of the data that the joint angles of the right body parts cause the quantitative difference between subjects. This implies that applying TDA together with conventional analysis methods can investigate the data in more detail.

In this paper, we used only β_0 and β_1 as the topological invariant. However, there is a possibility that our dataset has higher dimensional information. Thus, we need to consider the case for β_2 . In addition, we need to consider the other filtration criterion such as witness complex, alpha complex. The statistical analysis for the MDS result of each joint angle in the point clouds is also needed.

References

- [1] Sagioglu, S., & Sinanc, D. (2013, May). Big data: A review. In *2013 international conference on collaboration technologies and systems (CTS)* (pp. 42-47). IEEE.
- [2] Alpaydin, E. (2020). *Introduction to machine learning*. Massachusetts, USA: MIT press.
- [3] Bengio, Y., Goodfellow, I., & Courville, A. (2017). *Deep learning* (Vol. 1). Massachusetts, USA: MIT press.
- [4] Han, J., Pei, J., & Kamber, M. (2011). *Data mining: concepts and techniques*. Elsevier.
- [5] Carlsson, G. (2009). Topology and data. *Bulletin of the American Mathematical Society*, 46(2), 255-308.
- [6] Mémoli, F. (2011, August). Metric structures on datasets: stability and classification of algorithms. In *International Conference on Computer Analysis of Images and Patterns* (pp. 1-33). Springer, Berlin, Heidelberg.
- [7] Zerzucha, P., & Walczak, B. (2012). Concept of (dis) similarity in data analysis. *TrAC Trends in Analytical Chemistry*, 38, 116-128.
- [8] Ghrist, R. (2008). Barcodes: the persistent topology of data. *Bulletin of the American Mathematical Society*, 45(1), 61-75.
- [9] Edelsbrunner, H., & Harer, J. (2008). Persistent homology-a survey. *Contemporary mathematics*, 453, 257-282.
- [10] Cohen-Steiner, D., Edelsbrunner, H., & Harer, J. (2007). Stability of persistence diagrams. *Discrete & computational geometry*, 37(1), 103-120.
- [11] Adams, H., & Tausz, A. (2015). Javaplex tutorial. Retrieved from <http://goo.gl/5uaRoQ>
- [12] Maria, C., Boissonnat, J. D., Glisse, M., & Yvinec, M. (2014, August). The Gudhi library: Simplicial complexes and persistent homology. In *International Congress on Mathematical Software* (pp. 167-174). Springer, Berlin, Heidelberg.
- [13] Chazal, F., & Michel, B. (2017). An introduction to Topological Data Analysis: fundamental and practical aspects for data scientists. *arXiv preprint arXiv:1710.04019*.
- [14] Fasy, B. T., Kim, J., Lecci, F., & Maria, C. (2014). Introduction to the R package TDA. *arXiv preprint arXiv:1411.1830*.

- [15] Otter, N., Porter, M. A., Tillmann, U., Grindrod, P., & Harrington, H. A. (2017). A roadmap for the computation of persistent homology. *EPJ Data Science*, 6(1), 17.
- [16] Ferri, M. (2017). Persistent topology for natural data analysis—A survey. In *Towards Integrative Machine Learning and Knowledge Extraction* (pp. 117-133). Springer, Cham.
- [17] Pun, C. S., Xia, K., & Lee, S. X. (2018). Persistent-Homology-based Machine Learning and its Applications--A Survey. *arXiv preprint arXiv:1811.00252*.
- [18] Lee, H., Chung, M. K., Kang, H., Kim, B. N., & Lee, D. S. (2011, March). Discriminative persistent homology of brain networks. In *2011 IEEE international symposium on biomedical imaging: from nano to macro* (pp. 841-844). IEEE.
- [19] Lee, H., Kang, H., Chung, M. K., Kim, B. N., & Lee, D. S. (2012). Persistent brain network homology from the perspective of dendrogram. *IEEE transactions on medical imaging*, 31(12), 2267-2277.
- [20] Carlsson, G., & Mémoli, F. (2010). Characterization, stability and convergence of hierarchical clustering methods. *The Journal of Machine Learning Research*, 11, 1425-1470.
- [21] Mémoli, F. (2008, June). Gromov-Hausdorff distances in Euclidean spaces. In *2008 IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops* (pp. 1-8). IEEE.
- [22] Borg, I., Groenen, P. J., & Mair, P. (2012). *Applied multidimensional scaling*. Springer Science & Business Media.
- [23] Cho, K. D., Lee, E. J., Seo, T. H., Kim, K. R., & Koo, J. Y. (2012). General Research; Visualization of Bottleneck Distances for Persistence Diagram. *응용통계연구*, 25(6), 1009-1018.
- [24] Gamble, J., & Heo, G. (2010). Exploring uses of persistent homology for statistical analysis of landmark-based shape data. *Journal of Multivariate Analysis*, 101(9), 2184-2199.
- [25] Delory, B. M., Li, M., Topp, C. N., & Lobet, G. (2018). archiDART v3. 0: A new data analysis pipeline allowing the topological analysis of plant root systems. *FI000Research*, 7.
- [26] Costa, J. P., & Škraba, P. (2015). A topological data analysis approach to the epidemiology of influenza. In *SIKDD15 Conference Proceedings*.
- [27] Hajij, M., Jonoska, N., Kukushkin, D., & Saito, M. (2018). Graph-based analysis for gene segment organization in a scrambled genome. *arXiv preprint arXiv:1801.05922*.
- [28] Zomorodian, A. (2005). *Topology for computing* (Vol. 16). Cambridge university press.

- [29] Zomorodian, A. (2010). Fast construction of the Vietoris-Rips complex. *Computers & Graphics*, 34(3), 263-271.
- [30] Edelsbrunner, H., & Harer, J. (2010). *Computational topology: an introduction*. American Mathematical Soc.
- [31] De Silva, V., & Ghrist, R. (2007). Coverage in sensor networks via persistent homology. *Algebraic & Geometric Topology*, 7(1), 339-358.
- [32] Hatcher, A. (2002). *Algebraic topology*. Cambridge university press.
- [33] Aktas, M. E., Akbas, E., & El Fatmaoui, A. (2019). Persistence homology of networks: methods and applications. *Applied Network Science*, 4(1), 61.
- [34] Moon, C., Giansiracusa, N., & Lazar, N. A. (2018). Persistence terrace for topological inference of point cloud data. *Journal of Computational and Graphical Statistics*, 27(3), 576-586.
- [35] Zomorodian, A., & Carlsson, G. (2005). Computing persistent homology. *Discrete & Computational Geometry*, 33(2), 249-274.
- [36] Chung, M. K., Lee, H., DiChristofano, A., Ombao, H., & Solo, V. (2019). Exact topological inference of the resting-state brain networks in twins. *Network Neuroscience*, 3(3), 674-694.
- [37] Lee, H., Chung, M. K., Kang, H., Kim, B. N., & Lee, D. S. (2011, September). Computing the shape of brain networks using graph filtration and Gromov-Hausdorff metric. In *International Conference on Medical Image Computing and Computer-Assisted Intervention* (pp. 302-309). Springer, Berlin, Heidelberg.
- [38] Wickelmaier, F. (2003). An introduction to MDS. *Sound Quality Research Unit, Aalborg University, Denmark*, 46(5), 1-26.
- [39] Sammon, J. W. (1969). A nonlinear mapping for data structure analysis. *IEEE Transactions on computers*, 100(5), 401-409.
- [40] Chen, J., Ng, Y. K., Lin, L., Jiang, Y., & Li, S. (2019). On triangular Inequalities of correlation-based distances for gene expression profiles. *bioRxiv*, 582106.