MOX-Report No. 46/2019

funBI: a Biclustering Algorithm for Functional Data

Di Iorio, J.; Vantini, S.

MOX, Dipartimento di Matematica
Politecnico di Milano, Via Bonardi 9 - 20133 Milano (Italy)

mox-dmat@polimi.it http://mox.polimi.it

*fun*BI: a Biclustering Algorithm for Functional Data

Jacopo Di Iorio
MOX - Department of Mathematics, Politecnico di Milano
Simone Vantini
MOX - Department of Mathematics, Politecnico di Milano

December 2, 2019

Declaration of interest: none

Abstract

In order to group objects, a wide literature of methods, the majority of them known as clustering and biclustering methods, was created. In the meanwhile, the scientific community tried to defy the curse of dimensionality, dealing with problems characterized by data with one infinite continuous dimension: functional data. Even if many old and new clustering algorithms were generalized to these new types of data, biclustering methods did not share the same destiny. This paper fills the literature gap by defining the concept of bicluster for data described as a set of functions, and by introducing *fun*BI, the first biclustering algorithm that permits to find functional biclusters, i.e. subsets of functions that exhibit similar behaviour across the same continuous subsets of the domain. *fun*BI is a three-step algorithm based on DIANA, the most famous divisive hierarchical clustering method. The use of DIANA allows to visualize and to guide the searching procedure using dendrograms and cutting thresholds. Biclustering Clustering Functional data

1 Introduction

One of the fundamental need in data mining is to group a given set of objects according to some measure of similarity or dissimilarity. In order to do so the scientific community has created a wide number of algorithms and methods. The most famous of them are known under the name of clustering methods, usually applicable to data arranged in a data matrix. The main element of clustering is the similarity between rows or columns of the data matrix. This procedure leads to the discovery of some similarity groups at the expense of obscuring other similarity groups: it is not possible to have row groups and

column groups at the same time. Indeed, considering the two dimensions of data matrices, rows and columns, observations and features, one can obtain groups of similar observations according to all the features or, instead, groups of similar features according to all the observations. Furthermore, most of these algorithms seek a disjoint cover of the set of elements, i.e. they require that two cluster groups can not overlap and that each element, row or column, must be clustered into exactly one group. To overcome these limitations, a large number of algorithms that perform simultaneous and overlapping clustering on both the dimensions of the data matrix has been proposed under the name of biclustering, co-clustering, bi-dimensional clustering or subspace clustering (Pontes et al., 2015).

In the meanwhile, thanks to the augmented possibilities in collecting and storing data, researchers started to deal with problems described by data having a huge number of features. This is the case of functional data that are usually represented as a set of random variables taking values in an infinite dimensional functional space. In order to group them, the scientific community produced a flourishing literature about clustering approaches suitable for functional data (Jacques and Preda, 2014). Therefore, similarly to what it has been done with clustering, (e.g. the d_0 distance k-means by Tarpey and Kinateder, 2003 or the d_0 distance hierarchical clustering proposed by Ferraty and Vieu, 2006), it would sound natural to generalize biclustering methods to functional data by defining functional biclusters and by proposing algorithms able to detect them. However, this was not the case. Therefore, this paper fills this notable fap in the literature by giving a definition for functional biclusters and by proposing the *fun*BI algorithm, a three-step algorithm for functional data able to identify biclusters, subsets of functions that exhibit similar behaviour across the same continuous subset of the domain.

This article is structured as follows: Section 2 is the state of the art on biclustering for multivariate data and on functional clustering literature ; our proposal for the definition of a functional bicluster is presented in Section 3; Section 4 gives an insight on the parameters introduced to define a functional bicluster; in Section 5 the *fun*BI algorithm is presented; finally, in order to explain the practical usefulness of this new functional data analysis method, two case studies are presented in Section 6.

2 State of the Art on Biclustering for Multivariate Data and on Custering for Functional Data

Even if a very similar idea to modern biclustering was theoretically introduced by Hartigan (1972) in the 1970s, Cheng and Church (2000) are recognized as the first ones to propose a biclustering algorithm. Precisely, they developed this method to analyze gene expression data, to find subgroups of genes (i.e. columns) and subgroups of conditions (i.e. rows), where the genes exhibit highly

correlated activities for every condition, in order to understand the biological functions associated to each gene. Due to the popularity of this kind of data, building on this seminal paper, many new contributions were added to the list of biclustering techniques. This explains why new algorithms are currently still developed or old ones are improved, e.g. the recent biclustering based on PAttern Mining Software (BicPAMS) (Henriques et al., 2017) .

In general, there are many different ways to classify the wide range of multivariate biclustering techniques. For instance, the famous survey by Madeira and Oliveira (2004) proposes a biclustering taxonomy based on two dimensions: the structure and the type of identified biclusters. The structure taxonomy is about the number of results and overlapping strategies or, in general, in which way rows and columns are incorporated in biclusters. On the other hand, the type taxonomy aims at categorizing the results in terms of exhibited patterns and values.

Another classification is the one proposed by Pontes, Giraldez and Aguilar-Ruiz (2015). In their recent review, they propose to classify a large number of multivariate biclustering approaches existing in literature into two main categories: biclustering algorithms based on evaluation measures and non metric-based biclustering ones. In this paper, the latter classification is the one to be followed.

The first class of methods includes all those algorithms that are able to identify biclusters according to a defined evaluation measure. Representatives of this class are, for example, iterative greedy algorithms such as the Cheng and Church Biclustering algorithm based on the Mean Squared Residue (MSR), the Bimax (Prelić et al., 2006) and the Biclustering by Correlated and Large Number of Individual Clustered seeds (BICLIC) (Yun and Yi, 2013) which uses the Pearson correlation coefficient. In addition to these deterministic approaches, some authors have proposed to use stochastic strategies in order to add some randomness to the iterative greedy search they perform: Flexible Overlapped biClustering (FLOC) (Yang et al., 2003) and Random Walk Biclustering (RWB) (Angiulli et al., 2008) are two examples. There are also other procedures which base their search on the combined use of traditional one-dimension clustering and additional strategies in order to deal with the second dimension.

The second class of methods, instead, excludes the use of any evaluation measure to guide the search, preferring to represent data and biclusters in other ways. For instance, graph-based approaches, such as the Statistical-Algorithmic Method for Bicluster Analysis (SAMBA) (Tanay et al., 2002), use bipartite graphs or multi-graphs and then optimization techniques. Similar to the metric-based approaches aforementioned, there exists also a group of methods founded on the use of traditional one-dimension clustering algorithm and additional strategies to provide the second dimension analysis. It is important to notice that, differently from conceptually-similar first class methods, no evaluation measures are allowed. It is the case of the Coupled Two-Way Clustering (CTWC) (Getz et al., 2000). Plaid Models (Lazzeroni and Owen, 2002) and Conserved gene expression Motifs (xMotifs) (Murali and Kasif, 2002) are two of the most known approaches where biclusters are found thanks to the use of

probabilistic models. It is also possible to find biclusters using linear algebra. For example, Spectral Biclustering (Kluger et al., 2003) and the Iterative Signature Algorithm (ISA) (Bergmann et al., 2003), both based on Singular Value Decomposition (SVD), belong to this kind of methods.

Starting from 2005, many researchers, inspired by the overwhelming success of these techniques in the multivariate framework, developed new biclustering methods for time-series data, that are able to consider all the temporal relationships among elements. The first work dealing with time-series data was the Cheng and Church Time-Series Biclustering algorithm by Zhang et al. (2005), an algorithm based on the greedy procedure presented by Cheng and Church (2000) that is applied directly on the original data matrix. Instead, others methods, such as the one proposed by Ji and Tan (2004), the continuous coherent evolution Biclustering model (CCC-Biclustering) and its updated version both by Madeira and Oliveira (Madeira and Oliveira, 2007 and Madeira et al., 2010), and the k-CCC algorithm by Xue et al. (2014), work on a discretized version of the original matrix in different ways. This discretization is performed with sequence alignment and suffix trees without or with an error bound.

In recent years, thanks to the possibility of collecting and storing increasingly larger amount of information, the statistical community started to be interested in solving problems whose data were characterized by a very large dimension p. A special and extreme case of this type of data are functional data (i.e. data that can be represented as curves, surfaces,...). Functional datasets are modeled as samples of random variables which take values in an infinite dimensional functional space, e.g., a space of functions defined on some set T, for instance time interval. Functional data clustering received particular attention from statisticians in the last decade. According to their survey, Jacques and Preda (2014) categorize all the different clustering approaches into four groups: raw-data methods, filtering methods, adaptive methods, and distance-based methods.

The raw-data methods comprise all the techniques coming from the multivariate world. These methods cluster discretized ersions of functional data using multivariate clustering methods without reconstructing the functional form. These early approaches are unable to take into account the typical functional features such as continuity and derivatives.

In the filtering methods the high dimensionality of data is tamed by a filtering step which approximates the curves by means of a finite basis of expansions. After this first step, usually performed with B-splines or FPCA (see Ramsay and Silverman, 2007 for a general framework), multivariate clustering algorithms are used to define clusters of functional data. For instance, some algorithms propose k-means clustering on b-splines coefficients or principal component scores, while others apply unsupervised neural network to Gaussian coefficient's basis.

The adaptive methods collect contributes that consider the basis expansion as random variable having a cluster-specific probability distribution, instead that simple parameters. For this reason most of these methods are based on probabilistic modelling of basis expansion or of some FPCA scores.

Finally, distance-based methods try to adapt popular geometric clustering

algorithms, such as k-means and hierarchical clustering, to the functional setting. For these techniques it is necessary to define new specific distances or dissimilarities between functions. Depending on the definition and computation of these measures, the methods belonging to the last group can be also related to either raw-data or filtering methods.

Out of the taxonomy proposed by Jacques and Preda, one can find the recent sparse functional clustering methods that are capable of clustering the data while also selecting their most relevant features for classification. In Floriello et al. (2017) the functional sparse clustering is analytically defined as a variational problem with a hard thresholding constraint ensuring the sparsity of the solution. The ability to focus on subsets of the domain is a common characteristics between these algorithms and the one here proposed.

While researchers proposed a great number of clustering approaches suitable for functional data, the same fortune did not happen in the case of biclustering: precisely, to the best of our knowledge, the only works dedicated to such data are the proposals by Slimen et al. (2018) and Bouveyron et al. (2018). Their model-based procedures, taking inspiration from the multivariate model-based co-clustering world, are strongly funded on the latent block model and on its extension to the functional framework, called functional latent block model (funLBM). However, these two proposals deal with a particular type of functional data that can be described as a matrix whose entries are functions. Therefore, due to the presence of the matrix form, these two works could be interpreted as classical multivariate biclustering with the difference that the objects they deal with are functions and not numerical values. On the contrary, the procedure introduced in this paper deals with input data that a set of functions, with a single function corresponding to each observation. The difference between these two types of functional data is explained in Figure 1. Therefore the work here proposed analyses different problems and declines biclustering in a different ways from the aforementioned contributions. It is then possible to talk about two visions of functional biclustering: the one proposed by Slimen et al. and Bouveyron et al. performing a multivariate biclustering on matrices of functions, and the vision here explained. When the functional dataset presents a natural discrete structure, e.g. the curves of electrical use of different users day by day, the first strategies can easily define subgroups of users with similar electrical consumption in a subgroup of days. When there is no structure and the functional dataset is just a collection of functions, e.g. the curves of temperatures registered in different weather stations, the method here proposed can identify subgroups of stations having same temperature profiles in a defined interval of time. Depending to the problem, one strategy could be better suited than the other.

3 Definitions of Functional Bicluster

In the general multivariate biclustering framework, data are arranged in a matrix called $\mathbf{A}$ composed by n rows and m columns. $\mathbf{A}$ can also be expressed in

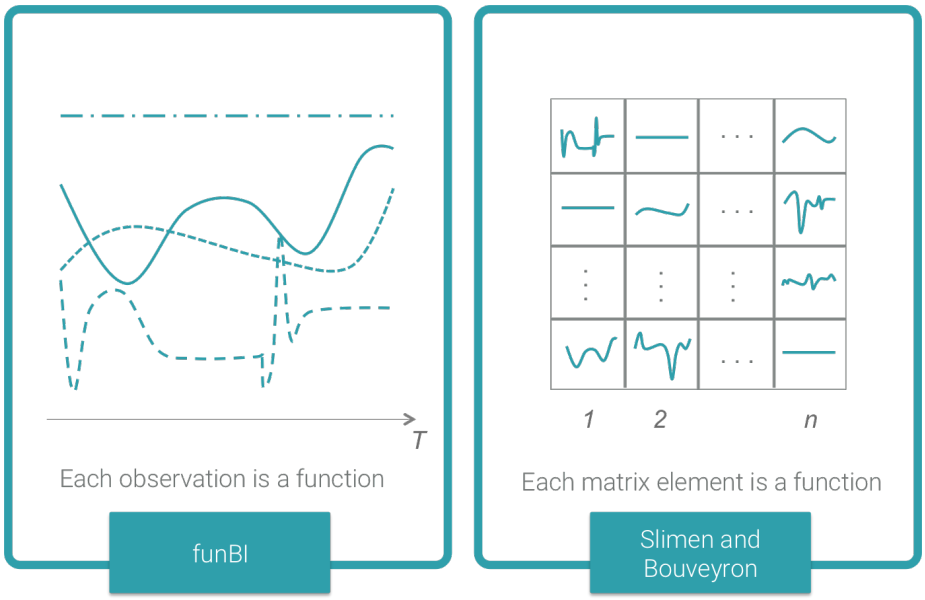

Figure 1: Functional data types used in this paper and in the works by Slime et al. and Bouveyron et al.

terms of its rows and columns as the couple $(\mathbf{X}, \mathbf{Y})$ where $\mathbf{X} = \{x_1, x_2, \ldots, x_n\}$ is its set of rows and $\mathbf{Y} = \{y_1, y_2, \ldots, y_m\}$ is its set of columns. It is then possible to identify any sub-matrix of the data matrix $\mathbf{A}$ as (I, J) where I and J are respectively subsets of $\mathbf{X}$ and $\mathbf{Y}$. Therefore (I, J) denotes the sub-matrix containing only the elements a_{ij} belonging to the sub-matrix with set of rows I and set of columns J.

Differently from clusters, which is a subset of rows (columns) that exhibit similar behaviour across all columns (rows), a multivariate bicluster is a subset of rows that exhibits similar behaviour across a subset of columns, and vice-versa. Therefore, it can be expressed as a sub-matrix (I, J) of the original data matrix $\mathbf{A}$ whose elements a_{ij} are similar to each others according to a defined evaluation measure. It is difficult to give a unique and perfectly fitting definition especially considering the importance of the measure used to define the result. However, as expressed in Madeira and Oliveira (2004) and in previous sections, biclusters can be categorized in terms of exhibited patterns.

The most natural version of ideal multivariate bicluster is the constant values bicluster: a submatrix (I, J) where all values are equal for all $i \in I$ and for all $j \in J$. This situation may be expressed by $a_{ij} = \mu$.

Considering the fact that there exists great practical interest in discovering if there are coherent variations on the rows or on the columns of the data matrix, the second type of ideal multivariate bicluster is the constant values on rows/columns one. In this case, for instance, a bicluster with constant values in

the rows identifies a subset of columns with the same behaviour across a subset of rows. This situation results in a sub-matrix (I, J) having all the values following the equation $a_{ij} = \mu + \alpha_i$ where μ is the mean value within the bicluster and α_i is an additive adjustment for row $i \in I$. The same formulations above explained can be also used to explain the ideal bicluster (I, J) with constant columns by substituting α_i with β_j, the additive adjustment for column $j \in J$. This leads to the identification of a subset of rows with the same behaviour across a subset of columns. It is important to notice how the duality of formulations is mandatory to express the fact that biclusters can follow a shifting pattern, represented by adding constant number (α_i or β_j). Visually representing rows as lines, shifting pattern gives a parallel (overlapping) behaviour among rows.

A more general case is represented by biclusters with coherent values both on rows and columns. A bicluster with coherent values both on rows and columns identifies a subset of rows showing a similar behaviour across a subset of columns up to a constant term and viceversa. Therefore, an ideal bicluster (I, J) with coherent values is defined as a subset of rows and a subset of columns, whose values are given by the expression $a_{ij} = \mu + \alpha_i + \beta_j$. Differently from the types of bicluster previously explained, there are two adjustments and not only one. By managing these adjustments it is possible to consider the two above situations as special cases of this last bicluster typology.

Moving to the functional framework, a revision of the concept of bicluster is mandatory. Indeed, from a theoretical point of view, when there is no matricial structure, it is not correct to talk about rows and columns as it has been done before in the multivariate setting. Functional data, at least in the $L^2(t) \cap C(t)$, as explained in Section 2, can be imagined as data matrix composed by rows and a continuous infinity of columns. This would leave space for applying the majority of biclustering algorithms existing in the multivariate literature to a discretised version of the functional set. However this would be a false start: shuffling the columns, as every multivariate biclustering method does, it is not possible and actually meaningless with functional data. Differently from the multivariate framework where the order of the columns/variables has generally no meaning, allowing us to shuffle them, in functional data this approach would virtually destroy the intrinsic ordered and smooth nature of data. Therefore, we define a functional bicluster as a subset of functions, or curves, that exhibits similar behaviour across the same continuous subset of the domain T. For instance, when T is a time interval, an ideal bicluster is a subset of functions showing the same behaviour across a defined time sub-interval belonging to the domain T. Consistently with the existing literature, the concept of similarity has a central role in defining biclusters.

Taking inspiration from the most comprehensive formulation of multivariate bicluster, the coherent values on both rows and columns bicluster, it is possible to give the following definition of ideal functional bicluster:

Definition 1. *The ideal functional bicluster is a subset I of functions paired*

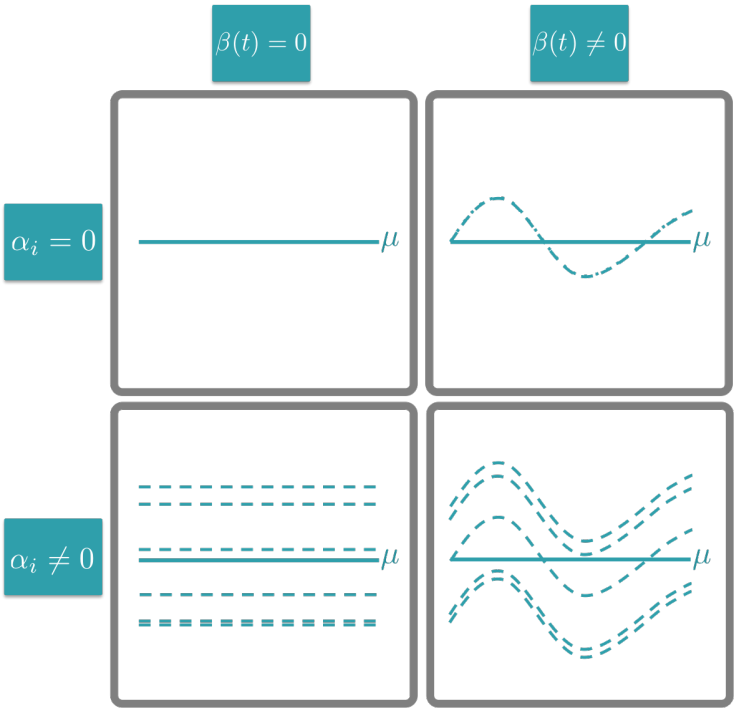

Figure 2: The four types of biclusters

with a sub-interval S of the domain T s.t.

$$f_i(t) = \mu + \alpha_i + \beta(t) \quad \forall i \in I \text{ and } t \in S, \tag{1}$$

where $f_i(t)$ is a general curve belonging to the bicluster, μ is the mean, α_i is the function-specific adjustment and $\beta(t)$ is the t-varying pattern of the bicluster.

To define uniquely the model parameters in order to make them identifiable, it is customary to impose the following constraints: $\sum_{i \in I} \alpha_i = 0$ and $\int_S \beta(t)dt = 0$.

Starting from Equation 1 one can define simpler kinds of biclusters. For instance, setting $\alpha_i = 0$ and $\beta(t) = 0$, the aforementioned expression is simplified in $f_i(t) = \mu$. This means that the bicluster is composed only by functions constantly equal to a given value μ on a sub-interval S of the time domain (top-left panel of Figure 2).

Setting $\alpha_i \neq 0$ and $\beta(t) = 0$, the obtained bicluster, expressed by $f_i(t) = \mu + \alpha_i$, is composed by parallel constant functions on S, sub-interval of T (bottom-left panel of Figure 2). On the other hand, setting $\beta(t) = 0$ instead of α_i, the found biclusters based on $f_i(t) = \mu + \beta(t)$, consists of functions behaving similarly as one can see in the top-right panel of Figure 2.

Therefore, the complete formulation expressed in Equation 1 identifies group

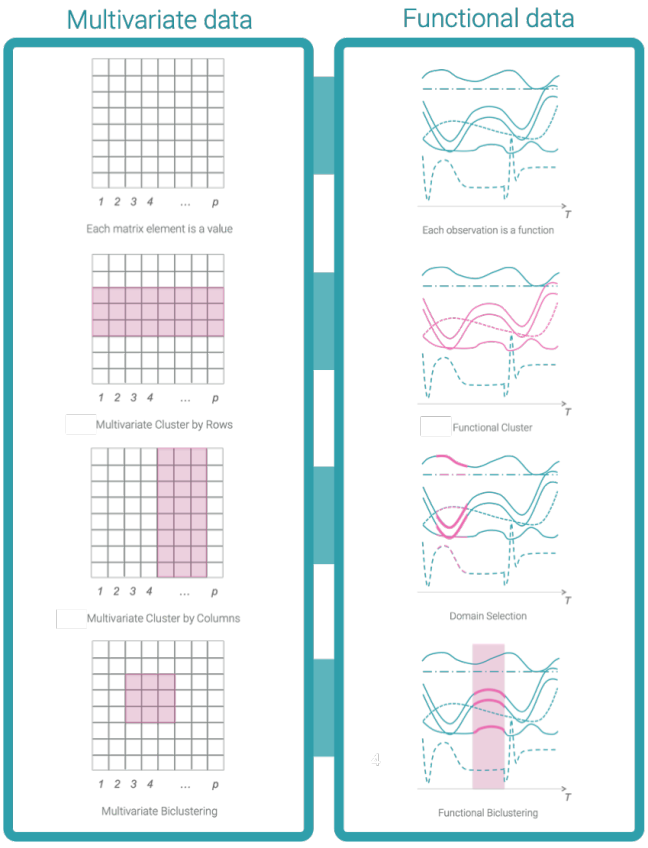

Figure 3: Differences in clustering and biclustering between multivariate and functional data.

of parallel non-necessarily constant functions on a sub-interval of the domain (bottom-right panel of Figure 2).

Therefore, coherently with the functional data considered in this paper and with our functional bicluster definition, the scheme in Figure 3 visually summarizes the main differences in data representation, clustering by one dimension, clustering by the other dimension and biclustering between the multivariate and the functional settings. Referring to the aforementioned Figure, it is important to consider how the nomenclature might change passing from one framework to another. While a multivariate cluster by rows (i.e., by observations) has its corresponding functional version in the functional cluster, the multivariate cluster by columns (i.e., by features) can be interpreted and named as domain selection in the functional setting. Instead, while the result of multivariate biclustering is a submatrix (produced after a rearrangement of the matrix itself), the result of functional biclustering is a selection of functions in a defined sub-interval,

as expressed by Equation 1 . However, this interpretation is not alien to the multivariate setting, where biclustering is often named subspace clustering.

4 Estimating the parameters of a functional bicluster

Real data usually present noise. The noise makes hard to identify perfect biclusters. For this reason, as already explained in Section 3, the concept of similarity has a central role in the definition of bicluster which is strongly related to the evaluation measure used. In the case of the seminal paper written by Cheng and Church (2000), the algorithm returns biclusters having the maximal dimension in terms of number of rows and columns according to the minimization of a score called the mean squared residue score. However, changing the framework from multivariate to functional, it is mandatory to coherently redefine the score used to validate biclusters. The mean squared residue score for a functional bicluster (I, S) can be written in accordance with the general functional additive model expressed in Equation 1, in the following way:

$$H(I,S) = \frac{1}{\mid I \mid} \frac{1}{\mid S \mid} \sum_{i \in I} \int_S (f_i(t) - (\hat{\mu} + \hat{\alpha}_i + \hat{\beta}(t)))^2. \tag{2}$$

In details, $f_i(t)$ is the value of function i at instant t, while $\hat{\mu}$, $\hat{\alpha}_i$ and $\hat{\beta}(t)$, the estimates of μ, α_i and $\beta(t)$, are respectively defined as:

$$\hat{\mu} = f_{IS} = \frac{1}{\mid I \mid} \frac{1}{\mid S \mid} \sum_{i \in I} \int_S f_i(t) dt \tag{3}$$

$$\hat{\alpha}_i = f_{iS} - \hat{\mu} = \frac{1}{\mid S \mid} \int_S f_i(t) dt \ \ - \hat{\mu} \tag{4}$$

$$\hat{\beta}(t) = f_{I(t)} - \hat{\mu} = \frac{1}{\mid I \mid} \sum_{i \in I} f_i(t) \ \ - \hat{\mu} \tag{5}$$

where f_{iS} is the integral mean of the function i in the sub-interval S, $f_{I(t)}$ is the sample mean of all the functions in I at the time instant t and f_{IS} is the general mean of all the curves in I in the whole sub-interval S (sample mean of the integral means).

Considering the relationships between $\hat{\mu}$, $\hat{\alpha}_i$, $\hat{\beta}(t)$ and $f_{I(t)}$, f_{IS}, f_{iS}, it is possible to write Equation 2 as follows:

$$H(I,S) = \frac{1}{\mid I \mid} \frac{1}{\mid S \mid} \sum_{i \in I} \int_S (f_i(t) - f_{iS} - f_{I(t)} + f_{IS})^2. \tag{6}$$

The mean squared residue score for functional biclusters is a measure of coherence used to validate biclusters. The optimum is given by the lowest score $H(I, S) = 0$, a situation that is visually represented by perfect parallel curves.

Modifying the model expressed in Equation 1 obliges to also modify the mean squared score for functional bicluster. Setting $\alpha = 0$, $\beta = 0$ or both, the changes are trivial and, in all the cases, the optimum is still represented by the lowest score $H(I, S) = 0$. Figure 2 shows the resulting perfect biclusters in all the possible situations.

5 Looking for a functional bicluster

Using the mean squared residue score for functional biclustering it is possible to estimate the parameters of a functional bicluster and to validate it. However, the main issue is to find the bicluster within the data. Many clustering and biclustering algorithms in the multivariate setting had to face the same problem that can be solved by using a brute-force approach. This approach is NP-hard in the worst case, but easily parallelizable.

Similarly, *fun*BI, the algorithm here proposed, follows a brute-force strategy when dealing with the continuous dimension. Precisely, it is a three-step iterative procedure (Figure 4). Each step has to perform a particular task.

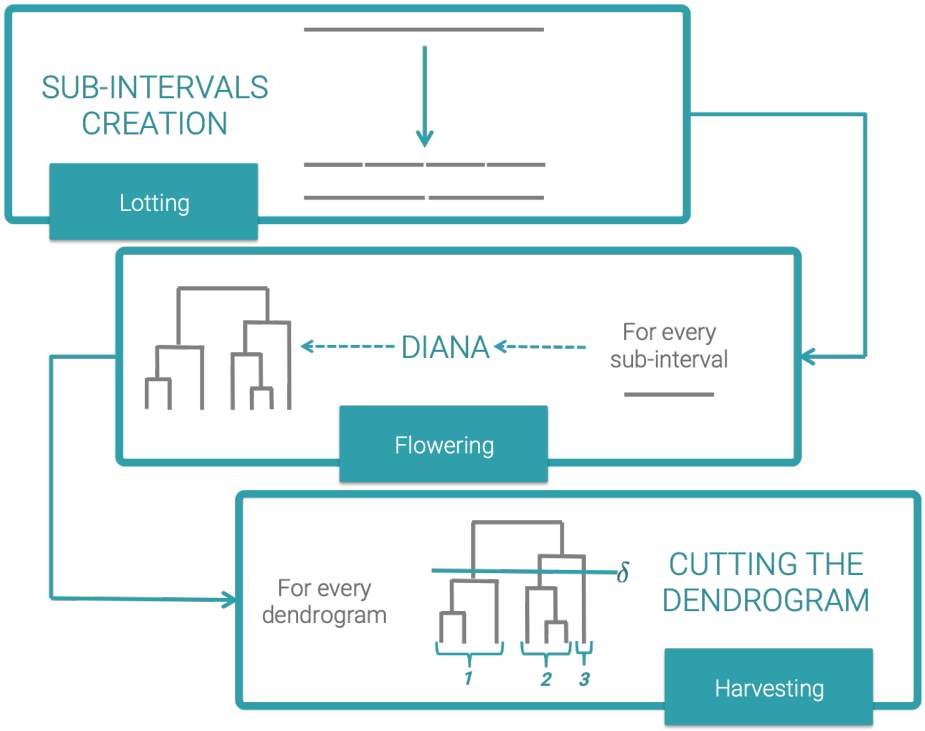

Figure 4: Synthesis of the proposed algorithm

The first step, named *Lotting*, aims at dividing the whole time interval T in sub-intervals S_w. It permits to reduce the computational complexity of taking

into account all the possible sub-intervals of the domain T. In the second step, called *Flowering*, a hierarchical clustering algorithm based on the mean squared score is performed on every sub-interval S_w in order to get a hierarchy of candidate biclusters. In the *Harvesting* step, the last one, all the candidate biclusters are collected. We propose the following routine.

Step 1 - Lotting Starting from a discrete grid based on the continuous dimension, the user decides the minimum length c of the continuous interval to analyze. Due to the computational complexity of considering all the possible sub-intervals, the *Lotting* procedure works as follows. Firstly, the continuous dimension is uniformly split into non-overlapping sub-intervals whose length is c. This creates the sub-intervals $S_w \in T \ :\mid S_w \mid = c$. Then all the sub-intervals $S_w \ :\mid S_w \mid > c$ are obtained by enlarging the first sub-intervals by multiples of c until covering the whole domain T.

Step 2 - Flowering Focusing on a defined sub-interval S_w of the domain T a mean squared score based DIANA (DIvisive ANAlysis Clustering) algorithm is performed Kaufman and Rousseeuw, 2009. The DIANA algorithm is the most famous divisive hierarchical clustering algorithm. Initially all data are in one cluster; an iterative procedure is then used to split the largest cluster in two parts, until each cluster contains only a single observation. The splitting procedure is performed in the following way: DIANA chooses the cluster with largest diameter, i.e., the maximum average dissimilarity; it initiates a new group called "splinter group" with the observation having the largest average dissimilarity from the other ones of the selected cluster; it reassigns to the splinter group all observations that are more similar to the new cluster than to the original one. The result is a division of the selected cluster into two new clusters. This procedure is then performed until each group counts only one observation.

In our case the dissimilarity matrix used by DIANA is based on the H_{score}. For each couple of curves $i, j \in I$, the dissimilarity d_{ij} between i and j is $H(\{i, j\}, S_w)$, i.e., the H_{score} of $(\{i, j\}, S_w)$, that is the matrix composed only by the two curves i and j evaluated in S_w:

$$d_{ij} = H(\{i, j\}, S_w). \tag{7}$$

Therefore, $d_{ij} = 0$ means that $H(\{i, j\}, S_w) = 0$ i.e. the curves i and j together compose a perfect bicluster in accordance to the additive model one is using.

The result of this step is presented in a dendrogram that synthesizes the DIANA top-down splitting procedure. Considering the fact that the dissimilarity matrix is based on the mean squared score, the height of the dendrogram used in the visualization is the H_{score}. The procedure aforementioned is performed for every sub-intervals S_w.

Step 3 - Harvesting All the candidate biclusters are collected. The collection can be perfomed according to two different strategies: the first one is user-driven and it needs a threshold for the H_{score} in a δ-biclustering fashion, while the second one is automatic and it is based on the gap statistic (Tibshirani et al., 2001).

According to the first strategy, the user decides a threshold for the H_{score} called δ-threshold. The value δ is then used to cut the dendrograms and, consequently, to identify δ-biclusters, i.e., biclusters having $H_{score} \leq \delta$. Cutting dendrograms in this way can generate a different number of biclusters for each sub-interval S_w.

The second strategy does not require any user intervention and is based on the automatic estimation of the number of clusters by the gap statistic. Using the output of the *Flowering* procedure, it compares the change in within-cluster dispersion with the one expected under an appropriate reference null distribution. However, in a multivariate situation we will not be able to choose a generally applicable and useful reference distribution: the geometry of the particular null distribution matters. Therefore, the main idea is to exploit the shape information in the principal components instead of using the MLE. The reference null distribution is then selected: (a) by generating each reference feature uniformly over the range of the observed values for that feature; (b) generate the reference features from a uniform distribution over a box aligned with the principal components of the data. In both cases the computation is possible thanks to a Monte Carlo sample drawn from the selected reference distribution.

For both strategies, it has been decided to use the H_{score} as height and vertical axis of the dendrogram, in order to have an immediate visual idea of the cutting result.

The three steps aforementioned, the *Lotting*, the *Flowering* and the *Harvesting* are extremely customizable. This makes the whole algorithm very flexible and easily modifiable based on the problem under exam.

However, after the *Harvesting* step a set of candidate biclusters is created. Even if it the best strategy would be to check them one by one with the advise of a problem domain expert, when the minimum length c is particularly small, this could be impossible because of the overwhelming number of results. Therefore, it could be useful to provide a way to order the results by importance and to reduce the number of biclusters: this is performed by the optional step of the *Tasting*. In order to analyze the main characteristics of the results, it has been decided to plot the resulting biclusters in a two axis plot as the one shown in Figure 5.

In this plot every dot is a bicluster whose sub-interval length is represented by the x-axis, numbers of functions by the y-axis and H_{score} by the dimension of the dot. In accordance with the existence multivariate literature, the most interesting biclusters are the ones having the lowest H_{score} and the maximum dimension (typically in terms of the number of rows and columns, while here in terms of the sub-interval lenght and the number of curves), although in

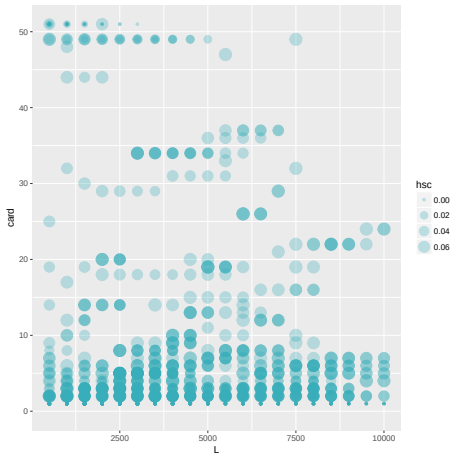

Figure 5: Plot of the resulting biclusters before reduction

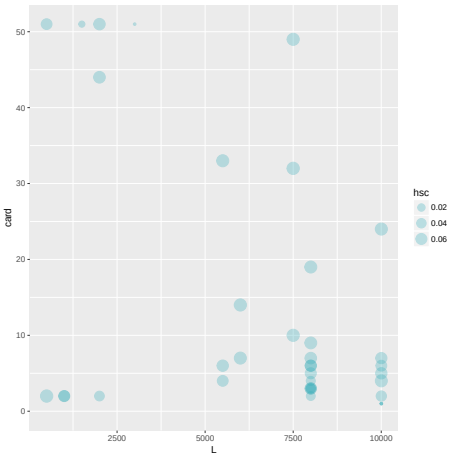

Figure 6: Plot of the resulting biclusters after reduction

some applications one could be interested in small and punctual phenomena considering low dimensions. Therefore, the most relevant results are usually the ones in the top-right corner whose dots are small.

In order to reduce even more the number of biclusters one can consider the fact that some of them are nested into some others. Given two biclusters (K, S_{w_1}) and (Y, S_{w_2}), if $K \subseteq Y$ and $S_{w_1} \subseteq S_{w_2}$, then the bicluster (K, S_{w_1}) can be removed since it is totally included in the bicluster (Y, S_{w_2}) and hence not informative. Figure 6 displays the resulting biclusters already presented in Figure 5 after performing the *Tasting* step. This strategy, of course, simplifies the revision of the results by reducing the number of the biclusters to minimal.

6 Case Studies

In this section two different case studies are presented, in order to highlight the practical usefulness of the introduction of *fun*BI to the functional data analysis.. Both the examples show how *fun*BI works and how it can detect important portions of the continuous domain leading to a segmentation of the functions that classical clustering techniques lack.

6.1 Case Study 1 - Aneurisk Project

The AneuRisk65 data have been collected within the AneuRisk project, a multi disciplinary scientific endeavour aiming at investigating the role of vessel morphology, blood fluid dynamics, and biomechanical properties of the vascular wall, on the pathogenesis of cerebral aneurysms. The data present, for each of the 65 subjects who took part to the project, both raw and preprocessed

information about the Inner Carotid Artery (ICA), described in terms of vessel centerline and radius profile. All data are available at `https://statistics.mox.polimi.it/aneurisk/`, the official website of the project. In this case study, according to the paper by Passerini et al. (2012), 50 z-first derivative curves of vessel centerlines after registration, defined in the last portion of the ICA and shown in Figure 7, are considered. The aim of this example is to high-

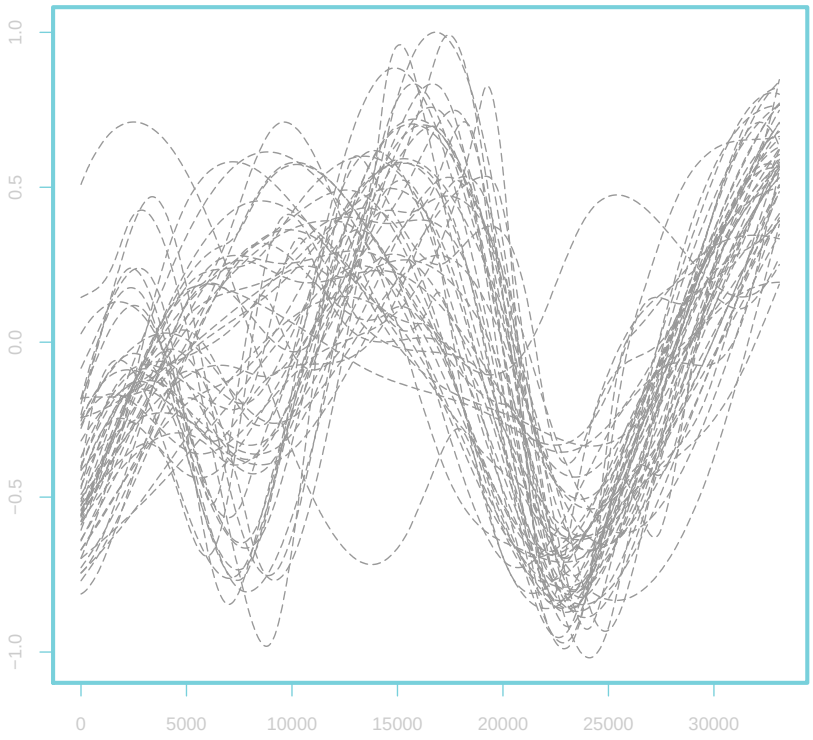

Figure 7: The 50 z-first derivative curves of the ICA vessel centerlines after registration

light how the functional biclustering algorithm here presented works, showing how it is able to identify grouping changes in the patter evolution of the considered functions. Precisely, *fun*BI can perform regular functional clustering, resulting in similar results to the ones obtained by other clustering procedures. Indeed, if in the *Lotting* step $c =\mid T \mid$ is chosen then the algorithm performs mean squared residue score-based DIANA to the whole domain.

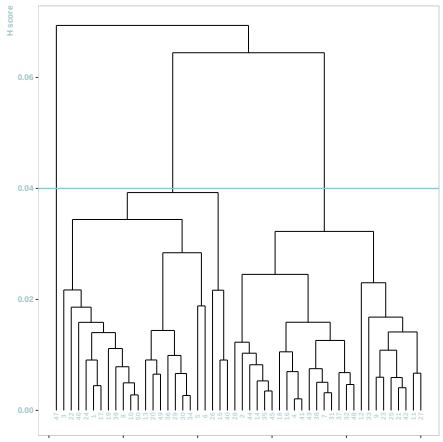

Figure 8: $\delta = 0.04$ cuts the dendrogram in 3 groups

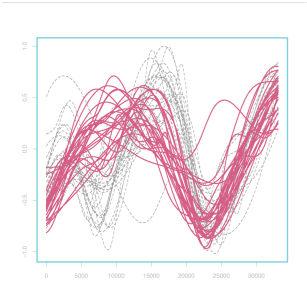

Figure 9: S shape vessels

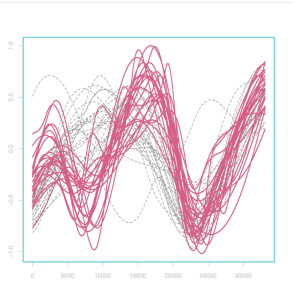

Figure 10: Ω shape vessels

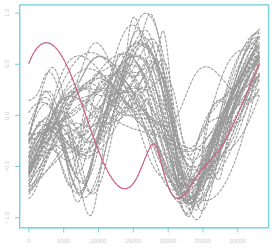

Figure 11: Single Outlier Group

Setting for instance $\delta = 0.04$, 3 biclusters/clusters are identified, as one can see from the dendrogram in Figure 8. These results identify the typical S and Ω shape groups of the vessel centerline of the inner carotid artery (Figure 9 and 10), two groups that are coherent with the medical literature (Huber, 1982) and previous works (Sangalli et al., 2009 and Sangalli et al., 2012), and a cluster composed by a single outlier (Figure 11).

However, the main scope of *fun*BI is to find functional biclusters in order to discover interesting dynamics that clustering is not able to highlight. Mantaining $\delta = 0.04$ but setting c to a value different than $\mid T \mid$, it is possible to obtain some useful new information. Precisely, in the case here presented, *fun*BI was able to detect 2212 biclusters, a great number of cases to consider. The *Tasting* procedure reduces the number of biclusters from 2212 to 158, performing a reduction of the 92.86%. Three biclusters which show the potential of the method are here reported. Figure 12 shows that in the considered sub-interval,

representing approximately the second half of the domain T, all the 50 functions, exception due for one, belong to the same group, meaning that the reason behind the identification of the S and Ω shape groups is not to be accounted for this particular portion of the vessels.

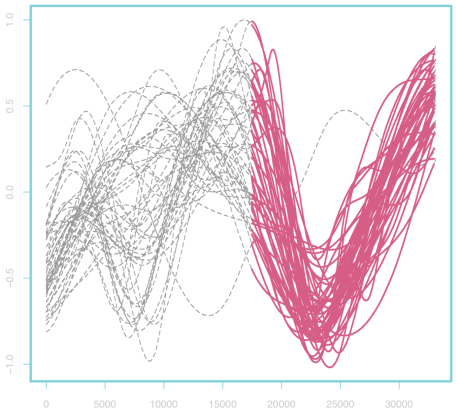

Figure 12: 49 z-first derivative curves of the ICA vessel centerlines composing one single bicluster

By focusing on the first half of T, the algorithm finds two biclusters (Figures 13 and 14) presenting two different shapes that might be connected to the results found by clustering: the first group present all the curves with a typical S-type vessels profile, while the second group is characterized by those curves with a Ω vessels profile.

6.2 Case Study 2 - Berkeley Growth Data

In this section, the results obtained by applying funBI on the growth curves included in the Berkeley Growth Study are illustrated. The dataset is one of the benchmark datasets for functional data analysis and it is also included in the R package fda (Ramsay et al., 2010). The study that collected these data, conducted by the California Institute of Child Welfare, is one of several long-term developmental investigations on children and it includes the heights (in cm) of 93 children, 54 girls and 39 boys, measured quarterly from 1 to 2 years, annually from 2 to 8 years and then biannually from 8 to 18 years. Therefore, even if the data are discrete observations, it is reasonable to consider them in the functional framework as realizations of a continuous process.

The purpose of this case study is to find out how *fun*BI can detect particular patterns defined in some intervals of the domain that the standard techniques cannot identify. Precisely, after preprocessing and reconstructing the curves monotonically (no alignment), one can notice that all the children, regardless of their sex, present the same feature, in particular a sharp peak of growth velocity between 10 and 16 year medically known as pubertal spurt. However

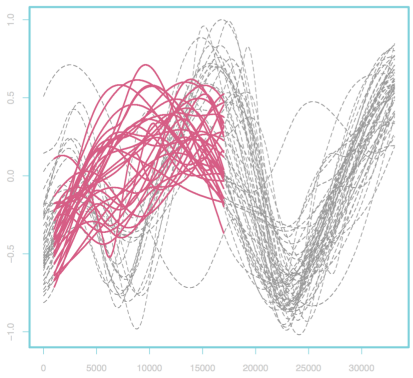

Figure 13: Group 1 in the first half of T

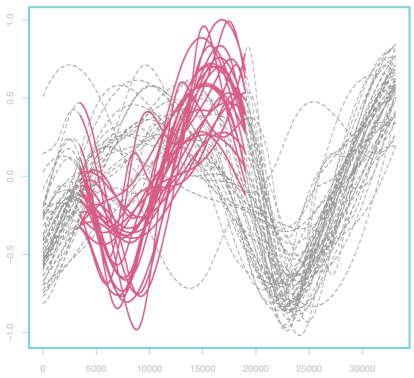

Figure 14: Group 2 in the first half of T

every children follows its own biological clock and this fact generates differences among the curves. Perfoming *fun*BI on the growth curves explains that the pubertal spurt is effectively the main source of differences among children. In order to do so, c, the minimum selected length of the sub-intervals, was set to cover the minimum gap among two registrations, and the *Harvesting* step was performed automatically using the gap statistic. In this way, the resulting biclusters displayed in Figure 15 shows that in general, without considering the whole pubertal spurt, but just what happens before and after this moment, there is only one unique pattern.

In addition, the main differences between girls and boys are connected with the pubertal spurt as shown in Figures 16, 17, 18 and 19 where the respective biclusters display a sexual partition based on different biological clocks.

In details, the bicluster presented in Figure 16 counts 22 functions, 19 boys and 3 girls, i.e., "girl25", "girl49" and "girl51". Therefore it is principally composed by those boys who experienced puberty before the others. The other male bicluster is displayed in Figure 17. It has 19 functions, 18 boys, who are the ones presenting the spurt after the others, and 1 girl, i.e., "girl33". Therefore, in these two biclusters there are two boys that are unaccounted for: "boy02" and "boy27" that both appear in the bicluster shown in Figure 18. It counts 23 observations, 2 boys ("boy02" and "boy27") and 21 girls having the pubertal spurt later then the others, that are all composing the remaining bicluster presented in Figure 19. Of course, as seen before, the two girls biclusters are missing of four girls belonging, instead, to the two boys groups.

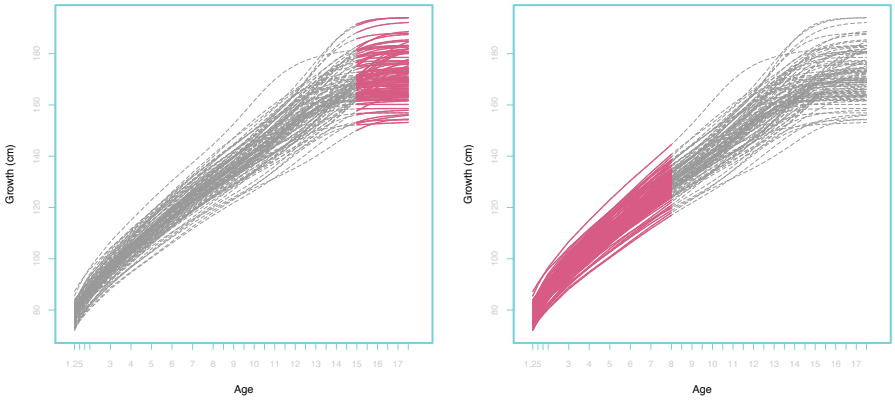

Figure 15: Considering what happens before or after the pubertal spurt, there is only one unique pattern

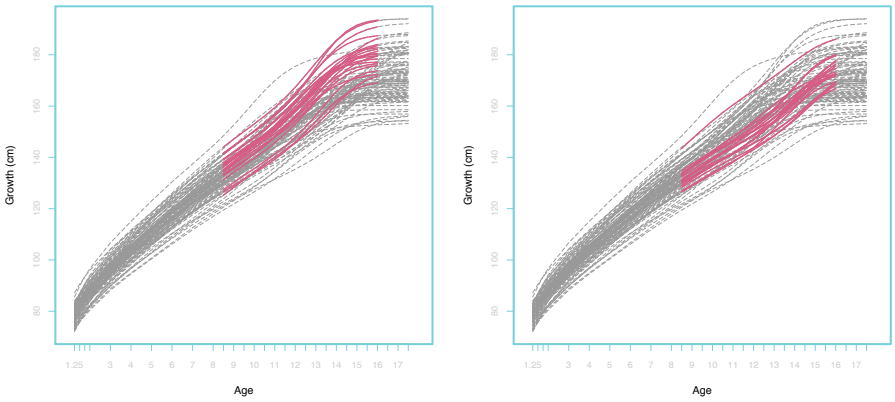

Figure 16: Boys bicluster 1

Figure 17: Boys bicluster 2

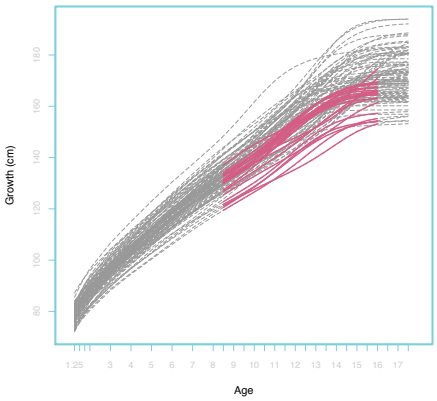

Figure 18: Girls bicluster 1

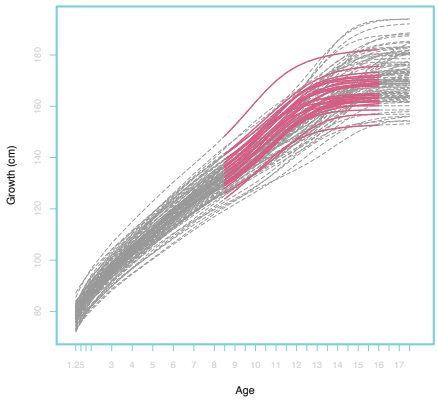

Figure 19: Girls bicluster 2

7 Discussion

This paper fills the literature gap concerning functional Biclustering by giving a first definition of bicluster for data whose observations are functions. Taking inspiration from the most comprehensive formulation of multivariate bicluster, the definition is based on an additive model that comprises, as special cases, easier kinds of biclusters.Then, being the definition strongly related to the used evaluation measure, the functional mean squared score, a measure of coherence in the validation of biclusters, was introduced in order to be used by *fun*BI. *fun*BI is three-step algorithm that uses a brute-force approach and it is characterized by an high flexibility. All the three steps are, indeed, easily generalizable and they are easy to be visually represented, due to the use of dendrograms. A package running the algorithm proposed in this paper is available at `https://github.com/JacopoDior/funBI`. Nonetheless, there are still open points, which we did not fully discuss in details since they are beyond the scope of the present paper.

The computational complexity to deal with all possible sub-intervals in case of larger domain T, forced us to develop a *Lotting* step that could be improved by employing computationally cheaper strategies, mandatory when performing extreme types of lotting such as the one considering all the possible combinations of equally long sub-intervals. Using this latter approach corresponds to give to our functional data a fictional matricial structure, identical to the one considered by Bouveyron et al. 2018 and by Slimen et al. 2018. In this way, all the resulting candidate biclusters will count functions belonging not only to the same sub-intervals but also to different ones.

The structure of *fun*BI could be modified not only in the *Flowering* step, where different hierarchical clustering algorithms can be used, but also in the *Harvesting*, where one can use different collection strategies. Focusing on the

Harvesting step, it is important to notice that the two proposed alternatives to "harvest" biclustes answer to two defined needs. Knowing that the interesting results are all those below a defined mean squared score, it is better to define a δ-threshold; instead, when there are no available information about the H_{score}, an automatic selection by gap statistic is recommended. However, tuning the parameter δ for a δ-threshold *Harvesting* strategy can be an issue, especially when there is lack of guidance from the domain expert or when H_{S_w}, the H_{score} of all the functions defined in a particular sub-interval, largely changes from one sub-interval S_w to another. The alternative of using the gap statistic can solve the aforementioned problem but, being automatic, it fails in finding all the smaller biclusters having very small H_{score}.

As already observed, it could be really difficult to identify the most interesting biclusters when the amount of candidates is overwhelming. For this reason the *Tasting* procedure was introduced. This step could be further improved by adding some more visual and interactive tools to help in the selection.

In addition, *fun*BI can only discover subsets of functions with similar behaviour across the same continuous subsets of the domain. Consequently, it can not find similarity across different sub-intervals. Therefore, the use of an alignment strategy could be a possible direction of generalization in order to detect similar functions across different sub-intervals.

References

Angiulli, F., E. Cesario, and C. Pizzuti (2008). "Random walk biclustering for microarray data". In: *Information Sciences* 178.6, pp. 1479–1497.

Bergmann, S., J. Ihmels, and N. Barkai (2003). "Iterative signature algorithm for the analysis of large-scale gene expression data". In: *Physical review E* 67.3, p. 031902.

Bouveyron, C., L. Bozzi, J. Jacques, and F.-X. Jollois (2018). "The functional latent block model for the co-clustering of electricity consumption curves". In: *Journal of the Royal Statistical Society: Series C (Applied Statistics)* 67.4, pp. 897–915.

Cheng, Y. and G. M. Church (2000). "Biclustering of expression data." In: *Ismb*. Vol. 8. 2000, pp. 93–103.

Ferraty, F. and P. Vieu (2006). *Nonparametric functional data analysis: theory and practice*. Springer Science & Business Media.

Floriello, D. and V. Vitelli (2017). "Sparse clustering of functional data". In: *Journal of Multivariate Analysis* 154, pp. 1–18.

Getz, G., E. Levine, and E. Domany (2000). "Coupled two-way clustering analysis of gene microarray data". In: *Proceedings of the National Academy of Sciences* 97.22, pp. 12079–12084.

Hartigan, J. A. (1972). "Direct clustering of a data matrix". In: *Journal of the american statistical association* 67.337, pp. 123–129.

Henriques, R., F. L. Ferreira, and S. C. Madeira (2017). "BicPAMS: software for biological data analysis with pattern-based biclustering". In: *BMC bioinformatics* 18.1, p. 82.

Huber, P. (1982). *Krayenbühl/Yaşargil cerebral angiography*. Georg Thieme Verlag.

Jacques, J. and C. Preda (2014). "Functional data clustering: a survey". In: *Advances in Data Analysis and Classification* 8.3, pp. 231–255.

Ji, L. and K.-L. Tan (2004). "Mining gene expression data for positive and negative co-regulated gene clusters". In: *Bioinformatics* 20.16, pp. 2711–2718.

Kaufman, L. and P. J. Rousseeuw (2009). *Finding groups in data: an introduction to cluster analysis*. Vol. 344. John Wiley & Sons.

Kluger, Y., R. Basri, J. T. Chang, and M. Gerstein (2003). "Spectral biclustering of microarray data: coclustering genes and conditions". In: *Genome research* 13.4, pp. 703–716.

Lazzeroni, L. and A. Owen (2002). "Plaid models for gene expression data". In: *Statistica sinica*, pp. 61–86.

Madeira, S. C. and A. L. Oliveira (2004). "Biclustering algorithms for biological data analysis: a survey". In: *IEEE/ACM Transactions on Computational Biology and Bioinformatics (TCBB)* 1.1, pp. 24–45.

— (2007). "An efficient biclustering algorithm for finding genes with similar patterns in time-series expression data". In: *Proceedings Of The 5th Asia-Pacific Bioinformatics Conference*. World Scientific, pp. 67–80.

Madeira, S. C., M. C. Teixeira, I. Sa-Correia, and A. L. Oliveira (2010). "Identification of regulatory modules in time series gene expression data using a linear time biclustering algorithm". In: *IEEE/ACM Transactions on Computational Biology and Bioinformatics (TCBB)* 7.1, pp. 153–165.

Murali, T. and S. Kasif (2002). "Extracting conserved gene expression motifs from gene expression data". In: *Biocomputing 2003*. World Scientific, pp. 77–88.

Passerini, T., L. M. Sangalli, S. Vantini, M. Piccinelli, S. Bacigaluppi, L. Antiga, E. Boccardi, P. Secchi, and A. Veneziani (2012). "An integrated statistical investigation of internal carotid arteries of patients affected by cerebral aneurysms". In: *Cardiovascular Engineering and Technology* 3.1, pp. 26–40.

Pontes, B., R. Giráldez, and J. S. Aguilar-Ruiz (2015). "Biclustering on expression data: A review". In: *Journal of biomedical informatics* 57, pp. 163–180.

Prelić, A., S. Bleuler, P. Zimmermann, A. Wille, P. Bühlmann, W. Gruissem, L. Hennig, L. Thiele, and E. Zitzler (2006). "A systematic comparison and evaluation of biclustering methods for gene expression data". In: *Bioinformatics* 22.9, pp. 1122–1129.

Ramsay, J. O. and B. W. Silverman (2007). *Applied functional data analysis: methods and case studies*. Springer.

Ramsay, J., H. Wickham, S. Graves, and G. Hooker (2010). *fda: Functional Data Analysis. R package version 2.2. 6*.

Sangalli, L. M., P. Secchi, S. Vantini, and A. Veneziani (2009). "A case study in exploratory functional data analysis: geometrical features of the internal

carotid artery". In: *Journal of the American Statistical Association* 104.485, pp. 37–48.

Sangalli, L. M., P. Secchi, S. Vantini, and V. Vitelli (2012). "Joint clustering and alignment of functional data: an application to vascular geometries". In: *Advanced Statistical Methods for the Analysis of Large Data-Sets*. Springer, pp. 33–43.

Slimen, Y. B., S. Allio, and J. Jacques (2018). "Model-based co-clustering for functional data". In: *Neurocomputing* 291, pp. 97–108.

Tanay, A., R. Sharan, and R. Shamir (2002). "Discovering statistically significant biclusters in gene expression data". In: *Bioinformatics* 18.suppl_1, S136–S144.

Tarpey, T. and K. K. Kinateder (2003). "Clustering functional data". In: *Journal of classification* 20.1, pp. 093–114.

Tibshirani, R., G. Walther, and T. Hastie (2001). "Estimating the number of clusters in a data set via the gap statistic". In: *Journal of the Royal Statistical Society: Series B (Statistical Methodology)* 63.2, pp. 411–423.

Xue, Y., Z. Liao, M. Li, J. Luo, X. Hu, G. Luo, and W.-S. Chen (2014). "A new biclustering algorithm for time-series gene expression data analysis". In: *Computational Intelligence and Security (CIS), 2014 Tenth International Conference on*. IEEE, pp. 268–272.

Yang, J., H. Wang, W. Wang, and P. Yu (2003). "Enhanced biclustering on expression data". In: *Bioinformatics and Bioengineering, 2003. Proceedings. Third IEEE Symposium on*. IEEE, pp. 321–327.

Yun, T. and G.-S. Yi (2013). "Biclustering for the comprehensive search of correlated gene expression patterns using clustered seed expansion". In: *BMC genomics* 14.1, p. 144.

Zhang, Y., H. Zha, and C.-H. Chu (2005). "A time-series biclustering algorithm for revealing co-regulated genes". In: *null*. IEEE, pp. 32–37.

MOX Technical Reports, last issues

Dipartimento di Matematica
Politecnico di Milano, Via Bonardi 9 - 20133 Milano (Italy)

44/2019 Formaggia, L.; Gatti, F.; Zonca, S.
An XFEM/DG approach for fluid-structure interaction problems with contact

45/2019 Regazzoni, F.; Dedè, L.; Quarteroni, A.
Active force generation in cardiac muscle cells: mathematical modeling and numerical simulation of the actin-myosin interaction

41/2019 Abbà, A.; Bonaventura, L.; Recanati, A.; Tugnoli, M.;
Dynamical p-adaptivity for LES of compressible flows in a high order DG framework

42/2019 Martino, A.; Guatteri, G.; Paganoni, A.M.
hmmhdd Package: Hidden Markov Model for High Dimensional Data

43/2019 Antonietti, P.F.; Mazzieri, I.; Migliorini, F.
A space-time discontinuous Galerkin method for the elastic wave equation

38/2019 Massi, M.C.; Ieva, F.; Gasperoni, F.; Paganoni, A.M.
Minority Class Feature Selection through Semi-Supervised Deep Sparse Autoencoders

40/2019 Lovato, I.; Pini, A.; Stamm, A.; Vantini, S.
Model-free two-sample test for network-valued data

39/2019 Lovato, I.; Pini, A.; Stamm, A.; Taquet, M.; Vantini, S.
Multiscale null hypothesis testing for network-valued data: analysis of brain networks of patients with autism

36/2019 Salvador, M.; Dede', L.; Quarteroni, A.
An intergrid transfer operator using radial basis functions with application to cardiac electromechanics

37/2019 Menafoglio, A.; Secchi, P.
O2S2: a new venue for computational geostatistics