POLITECNICO
MILANO 1863

Department of Mathematics
Doctoral Program in
Mathematical Models and Methods in Engineering

Functional data analysis for high dimensional and complex genomic data

Doctoral Dissertation of:
Alice Carla Luisa Parodi

Supervisor:
Prof. Piercesare Secchi
The Chair of the Doctoral Program:
Prof. Irene Maria Sabadini
Tutor:
Prof. Roberto Lucchetti

Year 2016 - XXIX cycle

ABSTRACT

In this work we aim to connect the advanced statistical techniques of Functional Data Analysis with the high dimensional and complex setting of genomic data.

Specifically, we adapt some statistical techniques to the needs of the biological community, both providing solutions to specific problems with innovative biological insights and developing efficient tools to make our research easily fruitful for the community. We present how the identification of phase and amplitude variations in functional data can be useful to define and classify ChIP-sequencing profiles or cognitive decline curves of patients affected by Alzheimer's disease.

Beside that, we present a new method, called FLAME, to deal with high dimensional and sparse Function-on-Scalar linear regression. We introduce FLAME both in its theoretical aspects and in its algorithmic implementation and then we present some applications related to the genomic area. FLAME has been used to analyze the influence of Single Nucleotide Polymorphisms to the longitudinal measurements of lung development of children affected by asthma and the influence of the stool and buccal microbiome in the growth of children affected by an overweight condition.

CONTENTS

INTRODUCTION

The recent development of advanced techniques to collect life sciences data requires the concurrent introduction of efficient statistical tools to extract meaningful information from huge amount of complex records. At this stage, the real challenge is to fully analyze the collected data, isolating the relevant information contained, without any a priori reduction of the complexity of the problem to exploit the state-of-the art statistical techniques. Then, statistics has a key role in developing efficient methodologies to deal with this new life-science community challenge.

The research we present in this work has the aim to present some real applied problems, collected from the biological environment of the -omics community, as well as some advanced statistical techniques (related to the functional data analysis) to efficiently analyze these data.

From early XX century, when the Genetic science was born, many innovations have been introduced: from Gregor Mendel and his heredity studies, through the Watson and Crick DNA double helix and the complete coding of the human genome, we are now in the era of Next Generation Sequencing. Next Generation Sequencing is a collection of new methods and laboratory techniques to analyze the genome, the transcriptome and the epigenome to identify not only the sequence of nucleotides our DNA is made by, but also the complex mechanism of codification of proteins and the complete nucleus environment where the DNA is immersed.

These new techniques, however, produce measurements which could be intrinsically longitudinal, since they vary in time or space, or which could affect the phenotypic expression of individuals in different ways with the passage of time. Therefore, in this work we introduce functional data techniques, to avoid an a priori simplification of the data to make multidimensional statistical techniques applicable. Moreover, we ensure to deal with the computational effort required to analyze such a huge amount of data and to allow the reproducibility of the analysis in different datasets from similar experiments: to support the statistics and biologist community we develop efficient `R` packages, mainly coded in `c++`, and upload them on public repositories.

The dissertation is split into two parts. In Part i the main focus is the isolation of the phase and amplitude variability, while in Part ii the main focus is the analysis of the influence of external factors on functional data.

Phase and Amplitude variability

When we analyze a functional data set, two different sources of variability could be present within data: a variability in the magnitude of the signal (amplitude variability) and a variability in the time-system of measurement (phase variability). The paired analysis of these two distinct sources of variation can make great advantages in the detection of meaningful aspects of the signals. In this part we present two case studies, one from the epigenetic area of ChIP-sequencing experiments and one related to the progression of Alzheimer's disease, in which phase variability plays a key role. We introduce two methods to identify and then remove the phase variability from the data and to analyze the remaining amplitude variation.

Functional Linear Models

Beside the internal sources of variation considered in the previous part, functional data may be influenced by external factors. For example, some genomic measurements may affect phenotypic expressions. Moreover, since the phenotypic response can be longitudinally measured on time we aim to isolate the genomic measurements that really influence the phenotype and how their influence varies with the passage of time. This is why in this part we develop a new method, called FLAME: to deal with high dimension sparse Function-on-Scalar regression. Beside presenting the method with its theoretical properties and computational details, to show FLAME efficiency we introduce two real case studies related respectively to the influence of Single Nucleotide Polymorphisms to the lung development and to the influence of the microbiome on the growth of children.

Part I

PHASE AND AMPLITUDE VARIABILITY

... the problem is that the curves exhibit two types of variability. Amplitude variability pertains to the sizes of particular features, ignoring their timings. Phase variability is variation in the timings of the features without considering their sizes.

Ramsay and Silverman (2005)

This is how Ramsay and Silverman define Phase and Amplitude variability. The idea of this distinction comes from the observation that, given a set of curves $f_i(x)$, the variability among them can be due to a variation in the ordinate f or in the abscissae x. In the typical example of Berkeley Growth Data of Jones and Bayley (1941), we can notice that a child can be shorter than an other in a certain time since he is overall shorter than the other or since he has a different speed of growth and then at that time he hasn't reached the height of the other yet. The aim of alignment techniques is, then, the identification of the variation in the f_i that is due to an abscissae variation, so that the quantification of the phase variability is possible. Once this source of variability is detected and removed, meaning that the abscissa grid of each f_i is transformed through an opportune warp, curves have the same time-scale and the remaining variability is the amplitude variability, only due to ordinate variations.

Different approaches have been defined to deal with the problem of identification of phase and amplitude variability, as presented in Marron et al. (2015) and Vantini (2012). From Sakoe and Chiba (1978) with the Dynamic Time Warping and the definition of piecewise-linear warping functions, through the landmark registration, which identifies the characteristic features of functions and transforms the abscissa of curves to make the feature coincide in time, to the more general definition of functional registration of Ramsay and Silverman (2005) and Srivastava and Klassen (2016). In this last contest warping functions can assume more general shapes and different metric are introduce to compare functions.
Here we focus on the registration with affine warping functions, as presented in Sangalli et al. (2010): the abscissae grid of the functions is adjusted with the application of shifts and dilations. Given two functions f_1 and f_2, the set of possible transformations for the abscissae (or set of warping functions) $\mathcal{W}$ and a distance measure d to compare the two curves, we look for the function $h \in \mathcal{W}$ s.t.

$$h = \arg\min_{s \in \mathcal{W}} d(f_1 \circ s, f_2).$$

h is, then, the function to be applied to the abscissae grid of f_1 to make the time time-scale of the two functions as similar as possible and it is computed minimizing the distance of the transformed f_1 from f_2. $f_1^R = f_1 \circ h$ is defined as the registered curve of f_1 to f_2 and now $d(f_1^R, f_2)$ does not contain any more the contribution of a phase variability, but quantifies only the amplitude variability. It has to be noticed that $\mathcal{W}$ and the metric d have to satisfy some coherence requirements; for example, $\mathcal{W}$ should be a convex space with a group structure with respect to the function composition $\circ$ and d and $\mathcal{W}$ have to be consistent: a simultaneous warping of two generic functions f_1, f_2 with the same h doesn't modify their distance $d(f_1, f_2) = d(f_1 \circ h, f_2 \circ h) \ \forall h \in \mathcal{W}$. Affine transformations and L^2 distances, for example satisfy these requirements.

In the next chapters two examples of real datasets where this distinction is crucial and reveals key aspects of the data are presented: in Chapter 1 a ChIP-sequencing experiment is illustrated; while in Chapter 2 the focus is the progression of Alzheimer's disease.

A ChIP-Seq experiment

ChIP-sequencing is a technique presented in Barski and Zhao (2009) and developed in order to analyze the epigenetic structure of the cells, where epigenetic is *the field of genomic sciences studying the molecular mechanisms through which cells can modify their expression without modifying the coding information of the DNA sequence, but varying, for example, the three-dimensional structure of the molecule or the expression level of genes* (Carey (2012)). Specifically, epigenetic focuses on the analysis of proteins surrounding DNA to detect their influence on the DNA expression. ChIP-sequencing, in particular, consists of the isolation of a specific protein in the nucleus of cells and in the analysis of its connection with the double helix of the DNA. The output of this technique is a measurement throughout the genome of a count indicating the presence of the protein in the region next to the genomic sequence. The human genome consist of $3,234,830$ basis and then the output of a single ChIP-seq experiment is a longitudinal measure on these millions of basis. Then, a functional data approach allows to consider the longitudinal measurement as a curve on the genome domain. Specific bioinformatics tools, like MACS proposed by Zhang et al. (2008), isolate the relevant regions of the genome where there is evidence of the presence of the protein. Then the curve throughout the genomic domain is reduced to a set of curves on specific regions of the genome. This set of curves can show both differences

in amplitude and phase, as inspected in Chapter 1.
Moreover we present a further scenario in which the analysis of phase variability can improve current biological techniques. It influences the initial step of the definition of the curve on the genome: specifically, the count on the genome is obtained as a sum of two separate measurements which need to be aligned before the sum. Then, the systematic definition of phase variability improves the current method which deals just with the identification of significant points on the two measurements.

This Chapter is within the Genomic Computing Project including the Statistics Group of Politecnico di Milano and the IEO-IIT research group and it is part of an already submitted paper. Codes are available on the `FunChIP` R/Bioconductor package (Parodi et al. (2016)), whose vignette is proposed in Appendix A. The package is build up to deal with ChIP-seq data, but it can be used also to analyze similar genomic structures like ChIP-exp experiments (Rhee and Pugh (2012)).

Heterogenety of cognitive decline in dementia

In the analysis of the progression of Alzheimer's disease we focus on the cognitive decline of patients. The objective is the identification of the initial point of the development of the disease (*time-zero*). It is well known, in fact, that Alzheimer's disease can reveal itself at different ages of the patients and often the first neurological visit does not correspond to the first appearance of cognitive decline. Both the identification of the starting point of the cognitive decline as a specific age or as the first neurological visit can be inaccurate. Then here we propose to exploit the alignment techniques to accurately identify the initial point of the cognitive decline. Moreover, the evolution of Alzheimer's disease can differ depending on the severity of the pathology and we want to propose a method that allows to consider different evolutions of the disease. The curves we measure are the results of the Mini-Mental State Examination (MMSE) proposed by Folstein et al. (1975): it is an indicator of the cognitive ability of patients during time. Allowing the possibility of different evolutions of the cognitive decline of patients and variations in the starting points of curves, we split patients in two groups which reveal faster and slower decline of the disease. Moreover we aim to identify some risk factors and biochemical markers to increase diagnostic accuracy for patients with a faster or slower decline. In this analysis we focus on two measurements: the presence of the Tau protein in the cerebrospinal fluid and the presence of a variant of the APOE gene. The level of Tau

reflects the neuronal and axonal degeneration or the possibly formation of neurofibrillary tangles (Andreasen et al. (2001)) and then the development of the Alzheimer's disease. In Tanzi and Bertram (2001), instead, the $\epsilon 4$ variant of the APOE gene is confirmed to be strongly related to the development of Alzheimer's disease. In our analysis we prove that with an efficient definition of the *time-zero* of the disease and a consequent efficient split of patients in slower and faster decliners, these two biomarkers become very significant.

This work is part of an already submitted work developed with Professor Steven Kiddle, Caroline Johnston, Chris Wallace and Richard Dobson form the Genetic and Developmental Psychiatry Center of King's College of London and Cambridge Institute of Public Health.

1

A CHIP-SEQ EXPERIMENT: PHASE AND AMPLITUDE VARIABILITY TO PREPROCESS AND ANALYZE CURVES ON THE GENOME

1.1 INTRODUCTION

Chromatin Immunoprecipitation followed by sequencing (ChIP-seq) generates local accumulations of sequencing reads on the genome, which correspond to specific protein-DNA interactions or chromatin modifications. In Figure 1.1 a graphical overview of a ChIP-seq experiment is presented. Specifically, the sequenced reads gather together at the genomic regions corresponding to the interactions sites and their accumulation can be identified with specific tools (Wilbanks and Facciotti (2010)) to isolate the enriched regions on the genome and define the correspondent profile of reads (*peaks*). Enriched regions are detected by considering the total area of the correspondent peak above a background signal, usually neglecting their shapes, which instead may convey additional biological information. Peaks, in fact, exhibit a variety of shapes: for example, Transcription Factor (TF) peaks usually display a gaussian-like profile, while some histone marks can show more elongated contours. However, even for a specific transcription factor, enriched regions may display differences in shapes, which are rarely taken into account when analysing ChIP-seq data (Guo et al. (2010), Zhang et al. (2011), Mendoza-Parra et al. (2013)).

See for example Figure 1.2 where some Genome Browser profiles showing different peak shapes are presented.

Notably, Cremona et al. (2015) classify the peaks of the human transcription factor GATA-1 with five indices summarizing their shapes, and link the clusters they obtain to specific biological insights: among others, to different regulatory complexes and different changes in gene expression. Nevertheless, the choice of the indices only provides a projected view of the peak

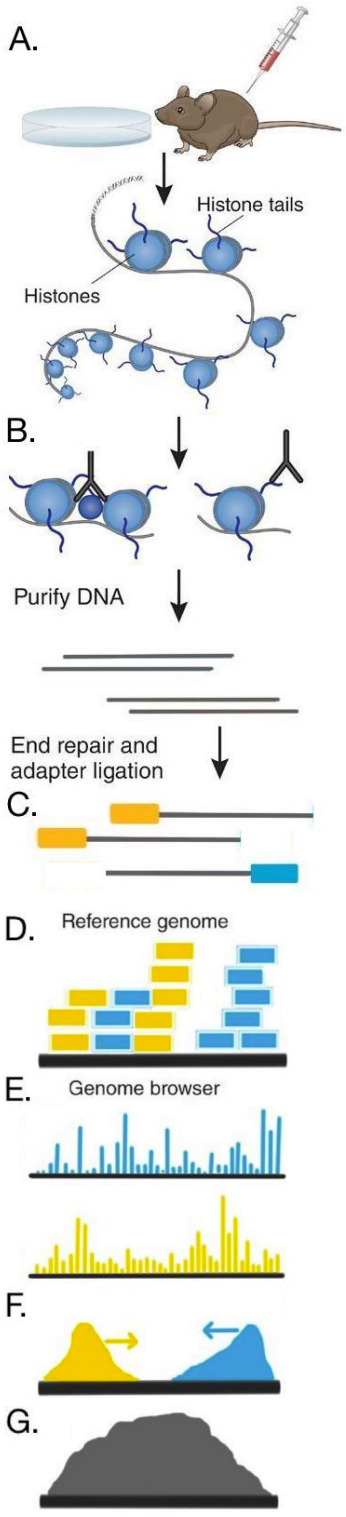


FIGURE 1.1: Figure from Kidder et al. (2011) to present the ChIP-sequencing technique. In the panel A. the experimental setting of the ChIP-seq experiment is introduced; histone marks and transcription factors which binds the DNA sequence are isolated through the linkage with a specific antibody (B.) and then the associated fragments of DNA are collected and purified by the transcription factor and the antibody. With specific tools the ends of the fragments (reads) are sequenced: in panel C. the right (called and 5′ end or negative read) and left (3′ end or positive read) reads are highlighted in blue and yellow: the coding sequence (i.e. the sequence of Adenine, Timine, Citosine and Guanine basis forming the segment) of the DNA is read. All the reads are then aligned on a reference genome and on each basis of the genome the number of positive and negative reads aligned there are counted, forming (E.) a sequence of positive and negative counts through the genome. Specific tools, like MACS (Wilbanks and Facciotti (2010)) identify enriched regions on the genome: these regions show counts statistically larger than the global count profile; there positive and negative counts are measured, the blue and yellow *peaks* of the panel F.. To define the global profile on the genome representing not only the positive and negative reads, but the entire fragments (gray lines of panel C.), it is necessary to estimate the fragment length. We propose a method to estimate this length considering the global shape of the positive and negative profiles. Finally (G.) for each region of the genome associated to the presence of the protein we can compute the a global count, summing the positive and negative counts, once the reads are extended up to the fragment length. It is the gray *peak* that we will analyze in its shape profile.

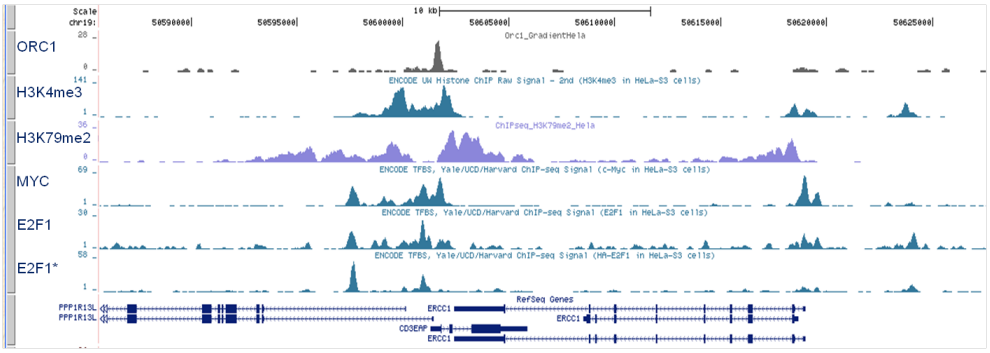


FIGURE 1.2: Genome Browser (Kent et al. (2002)) profiles of transcription factors and histone modifications showing different shapes both across different experiments and within the same experiment.

shapes along some specific axes, which may not capture all the relevant details. Here we present a method based on the isolation of phase and amplitude variability to define first of all the shapes of peaks and then to cluster them: after building up the peaks and approximating their profiles with cubic B-splines, the clustering technique classifies peaks applying a k-mean alignment and clustering algorithm. The method is implemented, together with all necessary preprocessing steps, in an user-friendly R/Bioconductor (Gentleman et al. (2004)) package `FunChIP` (Parodi et al. (2016)) which also provides some visualization tools for a quick inspection of the results. From the original set of aligned reads on the genome (yellow and blue segments of panel D. of Figure 1.1), stored in a `.bam` file, and the list of the enriched regions of the ChIP-seq experiment, in the `.bed` file, `FunChIP` builds up the positive and negative peaks of panel F., estimates the fragment length to build up the global profile of panel G., defines its functional approximation and classifies the global profiles on the enriched regions on the genome considering their shape. In this chapter we provide an application of this tool to some real ChIP-Seq datasets related to the transcription factor Myc in 3T9 murine fibroblasts; we show also how the clusters defined by the functional analysis are associated with different genomic locations and transcriptional regulatory activities.

This chapter is organized as follows. In Section 1.2 we outline the preprocessing steps applied to ChIP-seq data and present the clustering technique. In Section 1.3 we introduce the three datasets we will apply the clustering algorithm and in Section 1.4 the biological analyses we led. Finally, in Section 1.5 some concluding remarks.

In this work, we employ several functional data techniques to analyze ChIP-seq data. In particular, in Section 1.2.1 we describe how to estimate the fragment length l used in the sequencing, by aligning (shifting) the positive and negative reads generated in the ChIP-seq experiment. Then, in Section 1.2.2 we introduce an efficient algorithm, combining k-means clustering with a global alignment of the peaks, to classify the functional representations of ChIP-seq peaks.

1.2.1 *Preprocessing*

Given the location of the enriched regions on the genome, stored in a `.bed` file, and the `.bam` file containing the aligned reads of the ChIP-sequencing experiment, we introduce a method to define the global shape of the peaks identified by the experiment. As a first step, we collect for each genomic region i in $1, \dots N$, contained in the `.bed` file, the reads aligned on the positive and negative strands, and define the correspondent coverage functions, c_{i+} and c_{i-}. We assume that in each region the positive and negative coverages measure the same signal, shifted by a integer value d (see Figure 1.3 for a clarification). In our case of single-end sequencing, reads aligning on opposite strands are sequenced from the same fragments and the parameter d is then related to the original length of the fragments l as

$$l = d + r,$$

where r is the known read length.

The parameter d may be provided by the peak caller, as in case of MACS, but classically it is computed considering only the highest points of the positive and negative peaks, while we estimate d considering the whole profile of these peaks: we detect the shifting value which minimizes the global distance between the two groups of reads. In Supplementary Figure S1.1 we present the effects of setting $d = 0$ which corresponds to the definition of peaks as they are stored in the `.bam` file, ignoring the characteristics of the sequencing tools used in ChIP-Seq experiments. Focusing on the estimation of the d parameter, Figure 1.3 shows in the left panel the two peaks obtained with positive and negative coverage, while in the right panel the same two peaks, but shifted to minimize their distance. In particular, defining c^s_{i-} the

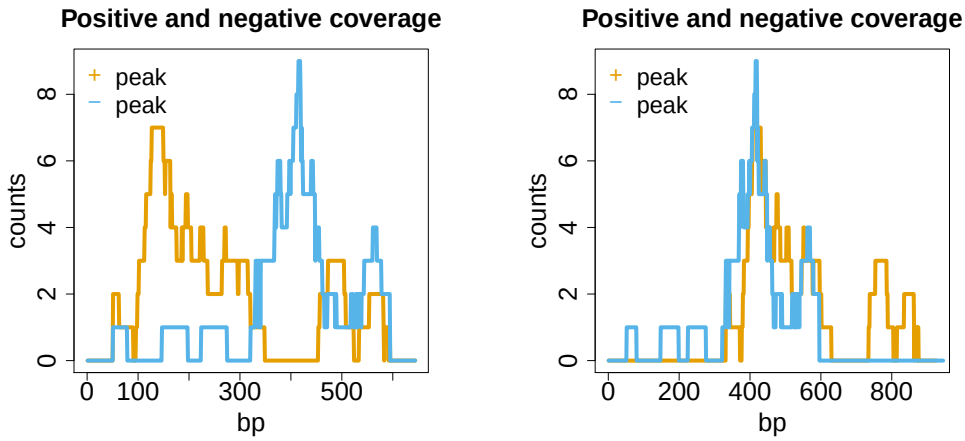


FIGURE 1.3: Left panel: positive and negative peaks in their original position. Right panel: the same positive and negative peaks, but shifted of the value d estimated from Equation (1.1).

negative coverage function of the ith region, shifted by s basis, d is computed as

$$d = \underset{s}{\operatorname{argmin}} \sum_{i=1}^{N} \mathcal{D}(c_{i+}, c_{i-}^{s}), \tag{1.1}$$

where $\mathcal{D}$ is a suitable distance between curves. In this specific case, given two functions $f(t)$ and $g(t)$ and the union of their domains Ω, $f^{\Omega}(t)$ and $g^{\Omega}(t)$ are $f(t)$ and $g(t)$ extended to zero where they are not defined on Ω. Then $\mathcal{D}$ is

$$\mathcal{D}(f, g) = \frac{\int_{\Omega} (f^{\Omega}(t) - g^{\Omega}(t))^2 dt}{|\Omega|}.$$

Once we have estimated the parameter d, we can compute for each region i the global coverage function c_i, obtained as the sum of c_{i-} and c_{i+}, extended on their 3′ ends up to the fragment length l. Figure 1.4 shows an example of peak obtained as global coverage, from the extension of positive and negative peaks, while in Figure 1.5 10 random peaks of the MycER0h dataset (Sabò et al. (2014)) are drawn. The peaks in Figure 1.5 are centered around their summits (or maximum point of the peak).

After computing the global coverage c for each enriched region of the `.bed` file, peaks are preprocessed to define the correspondent functions f and allow the application of functional data techniques. Here the description of the preprocessing steps required to define the functions f:

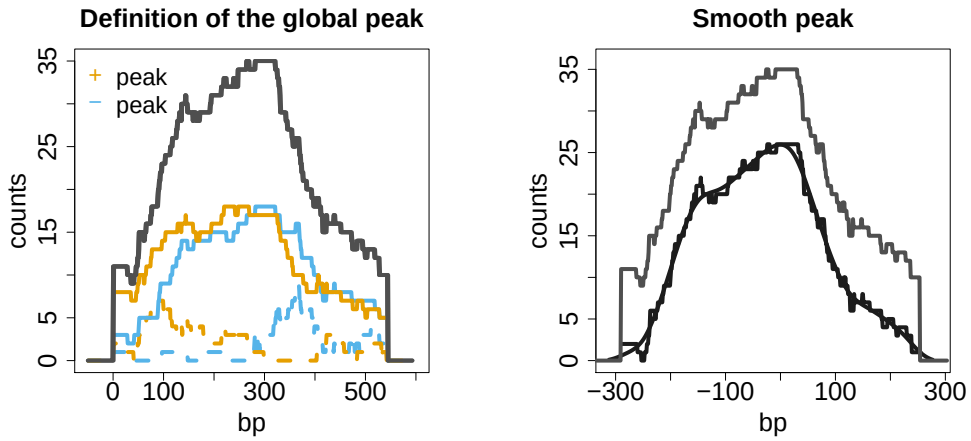


FIGURE 1.4: Global coverage function of a peak and smoothing. Left panel: Example of a peak (grey line), obtained as the sum of the positive (yellow) and negative (blue) reads, extended by d. The positive and negative peaks generated by the extended (continuous lines) and unextended (dashed lines) reads are also shown. Right panel: the original peak (grey line) and its representation after background removal and smoothing (black lines); peaks are centered around their summit.

1. *Removal of the background.* Given the characteristics of the ChIP-Seq experiments, each peak has a background, which can be generated by the specific sequence, PCR bias or random noise. In order to compare peaks, we estimate the background as constant along the peak and equal to the minimum value the count assumes, and we remove it from the data.

2. *Extension of the peak.* Each peak is defined on a specific enriched genomic region; we assume that, once we have removed the background, the peak is only a small part of the global coverage function on the whole genome that now assumes value 0 outside the enriched regions. Then each peak can be indefinitely extended with zeroes. This procedure allows the definition of a proper metric on the union of the domains to align peaks and isolate their phase and amplitude variability (see Section 1.2.2).

3. *Smoothing.* Peaks must be smoothed to allow the computation of the derivatives of the coverage functions. The smoothing is performed trough a cubic B-spline basis (Ramsay and Silverman (2005)), with knots every 50 nucleotides; this basis guarantees the continuity of functions and derivatives up to the second order. We introduce a

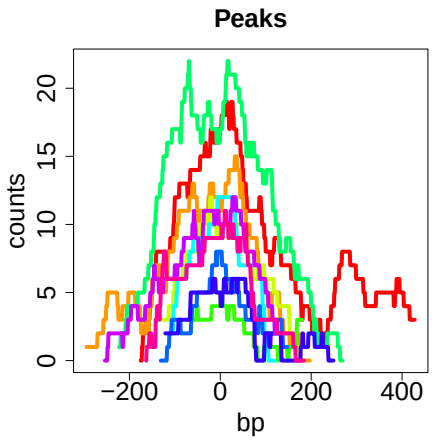


FIGURE 1.5: 10 randomly selected global peaks centered around their summits.

penalization on the second order derivative to control the roughness of the smoothed functions, measured in terms of changes of concavity. The smoothness parameter is estimated by minimizing the Generalized Cross Validation either on the data or on the derivatives. See Figure 1.4 (right) for an example of B-spline smoothing and Figure 1.6 (left panel) for the smooth approximation of the 10 random peaks previously introduced.

4. *Scaling.* A further optional preprocessing step consists in the scaling of the spline approximation. With this step we aim to isolate the shape of peaks, neglecting their width and area. All the peaks can be normalized to have the same width, equal to the minimum width of the peaks of the dataset, and area, equal to 1 (see Figure 1.6, right panel). The effects of scaling are detailed in Supplementary Figures S1.2, S1.3, S1.4, where the analyses we present on the MycER0h dataset are run on the scaled version of its peaks.

1.2.2 *Clustering: k-mean alignment method*

We adapt the k-mean alignment method, introduced in Bernardi et al. (2014); Patriarca et al. (2014); Sangalli et al. (2010, 2014) to ChIP-Seq peaks. The algorithm is an efficient method to perform unsupervised classification of functional data, taking into account their shapes and the possible data misalignment. A set of curves can be different either by amplitude (variability

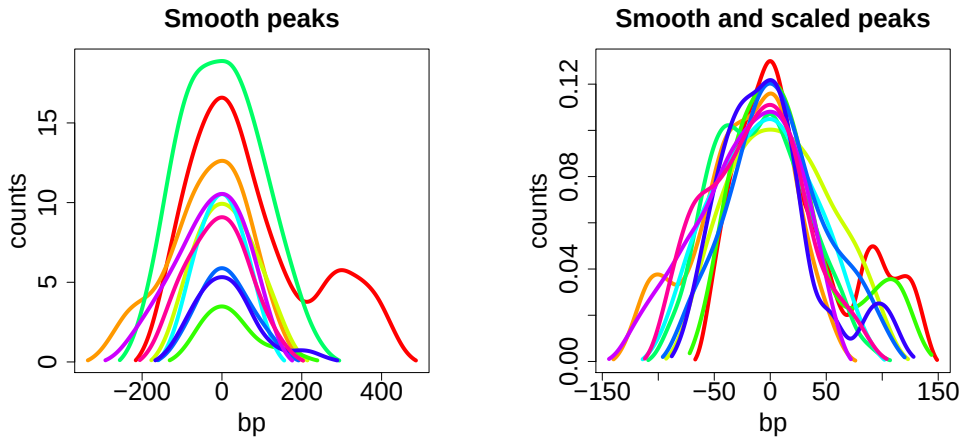


FIGURE 1.6: Preprocessing on the 10 peaks of Figure 1.5. Left panel: spline approximation of the peaks centered around their summits; right panel: the same peaks are scaled to the same area and width.

on the y axis) or by phase (variability on the x axis) (Marron et al. (2015); Ramsay and Silverman (2005); Vantini (2012)), and a classification method should take into account these two aspects together. The k-mean alignment algorithm (detailed in Algorithm 1) is an iterative procedure combining the k-mean classification method with the possibility of varying the phase of functions (alignment).

Two elements must be defined to run the algorithm, as explained in Sangalli et al. (2010): a class of warping functions $\mathcal{W}$ to define the alignment procedure, and a distance between two curves $\rho(\cdot,\cdot)$, together with a consistency requirement: simultaneous warping of two curves with the same warping function should not introduce a variation in their distance. The warping functions $\mathcal{W}$ should be a convex space with a group structure with respect to the function composition $\circ$ to ensure that nested applications give rise to functions of the same family.
In this work, we define $\mathcal{W}$ as the set of discrete shifts

$$\mathcal{W} = \{h : h(t) = t + q \text{ with } q \in \mathbb{Z}\}.$$

With this choice, two peaks can be shifted by an integer value to remove the phase variability. The distance function $\rho(\cdot,\cdot)$, instead, is a linear combination of the L^p distance of data and the L^p distance of derivatives:

$$\rho(f,g) = (1-\alpha)\|f-g\|_{L^p} + \alpha w \|f'-g'\|_{L^p}. \tag{1.2}$$

Algorithm 1: k-mean (k-medoid) alignment algorithm

Given a set of functions $f_1, \ldots, f_N$ and a number k of clusters

Template: random choice (if not provided) of the initial centers of the clusters $\tau_1 \ldots, \tau_k$

while *decrease of the distance* ρ *higher than a fixed threshold* **do**

  foreach $i \in 1 : N$ **do**

    Alignment: f_i is aligned to each template τ_j: the optimal warping function $h^\star_{i,j}$ in $\mathcal{W}$ is detected

$$h^\star_{i,j} = \underset{h \in \mathcal{W}}{\operatorname{argmin}} \ \rho(\tau_j, f_i \circ h)$$

    Assignment: f_i is assigned to the best cluster

$$j^\star_i = \underset{j \in 1:k}{\operatorname{argmin}} \ \rho(\tau_j, f_i \circ h^\star_{i,j})$$

  end

  foreach $j \in 1 : k$ **do**

    Template: identification of the new template of the cluster τ_j. In case of k-medoid algorithm, if $\{\tilde{f}_1, \ldots, \tilde{f}_{N_j}\}$ is the set of functions assigned to cluster j:

$$\tau_j = \underset{\tau \in \{\tilde{f}_1, \ldots, \tilde{f}_{N_j}\}}{\operatorname{argmin}} \sum_{i=1}^{N_j} \rho(\tau, \tilde{f}_i)$$

    Normalization: the average warping function of the curves belonging to j is set to be the identity transformation

  end

end

Specifically, here, we use the L^2 distance, where $\|f-g\|_{L^2}$ with f and g two functions is defined on the common domain Ω is

$$\|f-g\|_{L^2} = \sqrt{\int_\Omega (f(t)-g(t))^2 \, dt}.$$

Other possible choices for the distance are the L^1 distance or the L^∞ distance:

$$\|f-g\|_{L^1} = \int_\Omega |f(t)-g(t)|dt \quad \|f-g\|_{L^\infty} = \max_{t\in\Omega} |f(t)-g(t)|$$

Focusing on the weight α and w introduced in Equation 1.2, α is a user-defined parameter in $[0,1]$ and $w \in \mathbb{R}$ is chosen to balance the data and derivative contribution. In particular, we propose a definition of the weight w as the median of the ratio of the pairwise distances of data and derivatives:

$$w = \text{median}\left(\frac{\|f-g\|_{L^p}}{\|f'-g'\|_{L^p}}\right).$$

It is relevant to notice that f and g can be defined on domains with different length and in order to compute their distance f and g are extended with zeros on the union of their domains, as allowed by the preprocessing step of Section 1.2.1.

Finally, we note that at each step of the k-mean algorithm a center for each cluster must be defined (template). We choose it as the element of the cluster with minimum total distance from all the other members of the cluster (k-medoid algorithm).

1.2.3 *Clustering: definition of the final classification*

For a complete definition of the classification of the data with the algorithm of Section 1.2.2, we need to provide k, the number of clusters to split the dataset. We select this parameter in a data-driven way, analyzing different classifications obtained for different values of k. For each case, we compute the global distance within clusters, *i.e.* the sum on all clusters of the distance of each element of the cluster f_i from the correspondent template τ_j:

$$\rho_k = \sum_{j=1}^{k} \sum_{i=1}^{N_j} \rho(f_i, \tau_j),$$

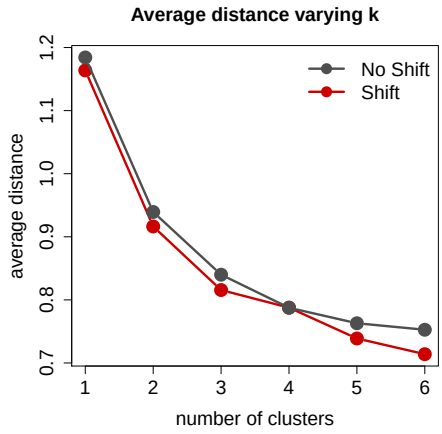


FIGURE 1.7: Global within-cluster distance d_k as a function of the number of clusters. Red line: peaks aligned to minimize distance; black line: unaligned peaks. This graph can be used to identify the best number of clusters for the classification of peaks and the contribution of the alignment procedure in the decrease of the global distance.

where N_j is the number of elements of the cluster j. A graphical representation of ρ_k/N, where N is the total number of peaks, as a function of k for the MycER0h dataset is shown in Figure 1.7. Clearly, the distance within a cluster decreases with k: the optimum value of this parameter is defined as the *elbow* of the curve, that is a value associated to a significant reduction of the distance when compared to the smaller values of k, and to a negligible variation when compared to higher values of k. In this plot we show both the result of the k-mean alignment algorithm and the result of the same classification, but without aligning peaks at each iteration. Clearly, the distance within clusters is reduced by aligning the peaks (red line, see Section 1.2.2): while in general aligning the peaks can introduce a sizeable decrease in the global within cluster distance, as shown for example in Supplementary Figure S1.2, in this case the effect is less pronounced, but still appreciable.
For this dataset, we consider the classification with alignment for $k = 2$ clusters; even $k = 3$ would have been a possible choice (see Supplementary Figures S1.5, S1.6, S1.7 and S1.8). However for $k = 3$, cluster 1 and 2 look very similar, made up by small and irregular peaks, and even in the biological analysis no strong difference between them is highlighted.

1.3 CASE STUDIES

We apply the method to three murine datasets, two obtained for the transcription factor Myc, and one for p53.

- MycER0h (Sabò et al. (2014)): murine 3T9 fibroblasts expressing a conditionally active Myc-oestrogen receptor chimera (MycER). This dataset displays endogeneous levels of Myc in exponentially growing cells. GEO accession number GSE51011, sample GSM1234508. Differentially regulated genes were obtained with respect to the activated cells (MycER4h): samples GSM1234745-GS1234748 (0h) and GSM1234749-GSM1234752 (4h).
- MycER4h (Sabò et al. (2014)): same cells as MycER0h. In the particular setting of fusion of Myc with the Estrogen Receptor (ER) that keeps the resulting chimeric protein in the cytoplasm, only upon 4-OHT administration, the chimera can rapidly translocate to the nucleus where Myc can exert its transcriptional activity. Then in the data collected 4 hours after the activation of the extra transgenic MycER construct (MycER4h dataset), the levels of Myc are much higher than in MycER0h, and the number of ChIP-Seq peaks is massively increased. GEO accession number GSE51011, sample GSM1234509.
- p53RAD (Tonelli et al. (2015)): murine B-cells exposed to whole-body ionizing radiation. The treatment causes DNA damage which in turn causes an activation of the transcription factor p53, which is present with high concentration in this sample. GEO accession number GSE71180, sample GSM1828856. Differentially regulated genes were obtained with respect to non-irradiated cells: samples GSM1828877-GSM1828880 (irradiated cells) and GSM1828869-GSM1828873 (non-irradiated cells).

1.4 SOME BIOLOGICAL INSIGHTS

We analyze the results of the classification to determine their biological significance. In particular, we focused on

- *Enrichment of peaks.* The enrichment of a peak is computed as

$$E = \log_2(n_p/N_p - n_I/N_I),$$

where n_p is the number of reads in the peak, N_p the total number of aligned reads in the experiment, n_I the number of reads in the peak

in the control sample, and N_I the total number of aligned reads in the control sample.

- *Genomic location of the peaks.* Each peak is annotated considering its overlap with promoters and genebodies: if a peak has at least a single nucleotide overlapping with a promoter region (defined as $[-2\text{kb}, 1\text{kb}]$ from the transcription start site, or TSS, except for the p53RAD dataset, where promoters are defined as $[-5\text{kb}, 2\text{kb}]$), it is classified as a promoter peak. Otherwise, if it overlaps with a gene body (defined as TSS+1kb to TES, or transcription end site) it is classified as intragenic. Finally, if it does not overlap with either feature, we consider a peak as intergenic. For these analyses, we used the mm9 assembly of the murine genome and the RefSeq annotation of genes.

- *Transcriptional regulation of genes*. For all the systems studied here, we downloaded RNA-Seq data in two conditions, characterized by different levels of the transcription factor of interest, and we integrated these data with ChIP-Seqs of the TF in both conditions. We analysed RNA-Seq data with the DESeq2 R/Bioconductor package (Love et al. (2014)), to identify genes whose promoter is bound by the TF of interest, and that are significantly changed between the two conditions (Benjamini-Hochberg adjusted p-value smaller than 0.05). Among these genes, we define as up-regulated those with a fold change greater than 1, and down-regulated those with fold change lower than -1. All the other expressed genes are termed non deregulated, or nodeg.

- *Motif analysis.* For each cluster, we performed an unsupervised motif discovery to detect motives enriched under the genomic regions covered by the correspondent peaks. Due to variations in the width of the peaks, we restricted these analyses to a ± 200bp region around the summit of each peak (± 100bp for p53RAD). We considered a random sample of 1000 peaks for each cluster, and we repeated the analysis 3 times. Then, we searched the position weight matrix obtained with motif discovery and associated to the TF of interest with the Biostrings R/Bioconductor package of Pagès et al. (2016) to find their positions in the sequences spanned by the peaks. The summits of the peaks were taken from the output of the peak caller MACS (Zhang et al. (2008)).

  In the case of Myc, we used DREME (Bailey (2011)) to discover short and ungapped motives enriched in the DNA sequences of the peaks, and we recovered its binding motif, called enhancer box, or E-box

(CACGTG, Figure 1.8A); we also found for the TGA-TCA motif (figure 1.8B) which has been previously associated to jun/fos binding (Gupta et al. (2007)).

(A) Myc binding motif (E-box)

(B) jun/fos binding motif

FIGURE 1.8: Myc binding motifs as estimated from the MycER0h dataset.

In the case of p53, we performed motif discovery with MEME (Bailey and Elkan (1994)) on the top 1000 enriched peaks in each cluster, as the binding motif of this TF is longer (see Figure 1.9) and less frequently found than in the case of Myc. We found that both cluster display the p53 binding motif, but with different significance.

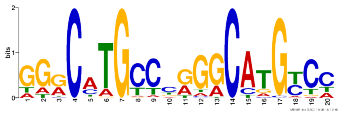


FIGURE 1.9: p53 binding motif as estimated from the p53RAD dataset

1.5 DISCUSSION

An example of the results is shown in Figure 1.10, in particular this is referred to the dataset related to the transcription factor Myc expressed at physiological levels in 3T9 mouse fibroblasts (MycER0h). Peaks are defined piling up the fragments obtained by the extension of the reads collected from the ChIP-Seq experiment with d estimated as Equation (1.1). As previously introduced, the global distance curve indicates $k = 2$ as a possible choice for the classification. Then, focusing on the composition of the clusters obtained, cluster 1 includes around 68% of the 15811 peaks of this dataset and it is mostly composed by small and irregular shapes, while in cluster 2 peaks are larger and more regular. This distinction reflects also a different distribution of the enrichment index of the peaks, with the p-value of the two-sided z test $< e10^{-16}$. Moreover, they are differently localized (see Figure 1.11), as confirmed by the χ^2 test for the differences in the locations in the two clusters, and in particular there is strong evidence to confirm a difference in the proportion of promoter peaks (χ^2 test p-value $< 10^{-16}$) in the two clus-

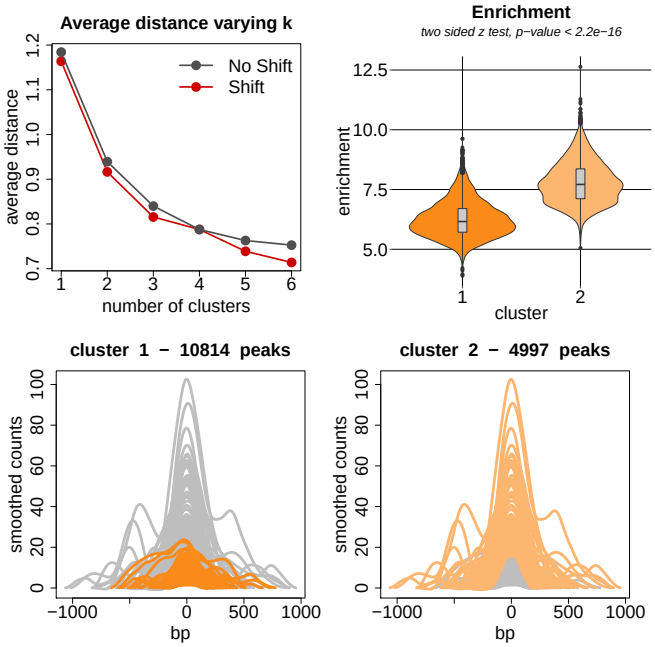


FIGURE 1.10: Classification of MycER0h peaks: Top left panel: distance within clusters, both with (red line) and without (black line) the alignment of peaks. Top right panel: the distribution of the enrichment of the peaks in the two clusters. Bottom panels: examples of peaks belonging to the two clusters, centered around their summits and aligned with the estimated shift coefficient.

ters. Peaks of the two clusters have also different associations with up and down regulated genes upon Myc overexpression (χ^2 test p-value $2 < 10^{-13}$). There is difference both for down-regulated genes (χ^2 test p-value $< 10^{-12}$) and for up-regulated genes, even if weaker (χ^2 test p-value $= 0.00043$). To conclude, focusing on the motif analysis presented in Figure 1.12, we show in the top panel the results of the motif discovery. In cluster 1, the main motif detected is TGAsTCA (E-significance between 10^{-34} and 10^{-27}). The Myc binding motif, or E-box, instead, is detected with a lower significance. In cluster 2, the same two motifs are found, but with inverted order of significance (E-value $< 10^{-82}$ for E-box, and between 10^{-26} and 10^{-22} for the TGAGTCA motif). This different prevalence of motifs is confirmed also by the supervised search of motives, with the E-box significantly more present in cluster 2, (χ^2 test p-value $< 10^{-16}$) and closer to the summit (z test p-value $< 10^{-16}$) with respect to cluster 1; the TGAsTCA motif, instead, is slightly more present in cluster 2, but equally distant from the summit of the peaks than in cluster 1.

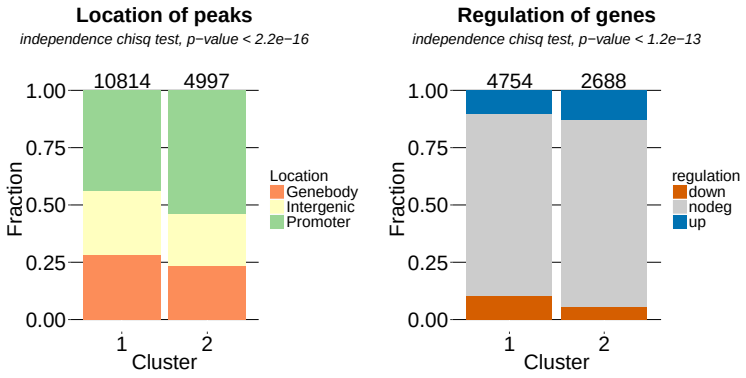


FIGURE 1.11: Genomic location of the MycER0h peaks distinguished in the two clusters. Left panel: genebody, intergenic and promoter distinction of peaks in the two clusters. Right panel: associations of promoter peaks with up and down regulated genes upon Myc overexpression.

Several further analyses on this dataset are proposed and fully described in Supplementary Figures. Briefly, we present

- a comparison with the multivariate clustering pipeline of (Cremona et al. (2015));
- the results of the functional classification of peaks obtained avoiding the extension of reads during the piling up procedure;
- the analysis of two other datasets (MycER4h and p53).

Specifically, applying the multivariate clustering pipeline (Cremona et al. (2015)) to MycER0h, we obtained again two clusters, greatly differing in enrichment (Supplementary Figures S1.9), as the area and height indices produce a classification strongly driven by size. In order to increase the focus on shape, one possibility would be to neglect these two indices in the classification: this would result in the identification of a cluster of small and seemingly regular peaks (Supplementary Figure S1.11). The idea of neglecting the area and the width is reflected in the functional algorithm in the scaling preprocessing step. In Supplementary Figure S1.2 the results of the classification of the MycER0h peaks, once they are scaled to the same area and width, are presented. Here we highlight the importance of the alignment: the red global distance line of the k-mean with alignment is far from the black line of the k-mean without alignment. With the distinction in $k = 3$ clusters, beside the regularity, we also detect differences in the symmetry of peaks. However, the analysis presented in Supplementary Figure S1.3 and S1.4 don't show

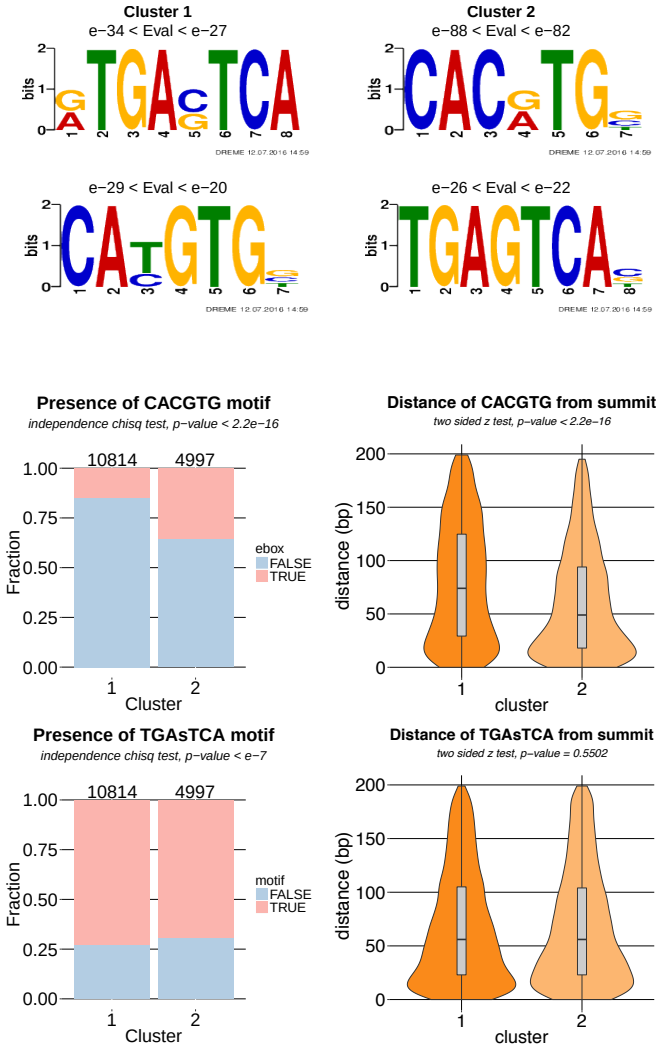


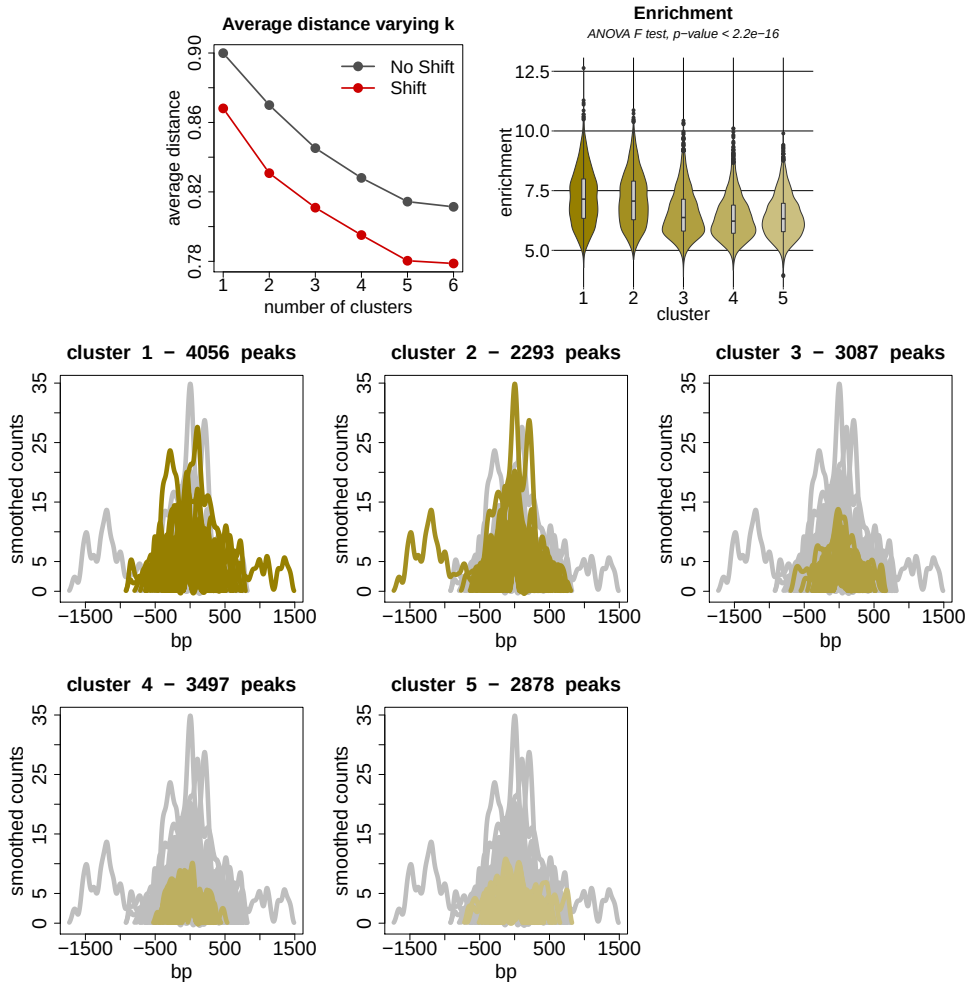
FIGURE 1.12: Motif analysis of of the MycER0h peaks distinguished in the two clusters. Top panels: results of the motif discovery performed on the peaks of the two clusters. Middle panels: presence of the E-box in the two clusters and distance from the summit of peaks. Bottom panels: presence of the TGAsTCA motif and distance from summit.

strong biological consequences of these differences in symmetry. In the two examples presented here, we globally conclude that the functional analysis is less related to the magnitude of the peaks and more focused on the shapes of data, compared to the multivariate classification. Specifically, focusing on the comparison of the classification via the 5 indices multivariate classification and the functional method on non-scaled peaks (Supplementary Figure S1.10) we notice how the functional classification is less related to the size of peaks. Comparing, instead, the 3 indices multivariate classification and the scaled peaks functional classification (Supplementary Figure S1.12), we detect that the functional classification can isolate fine details on shapes that are not captured by the multivariate indices.

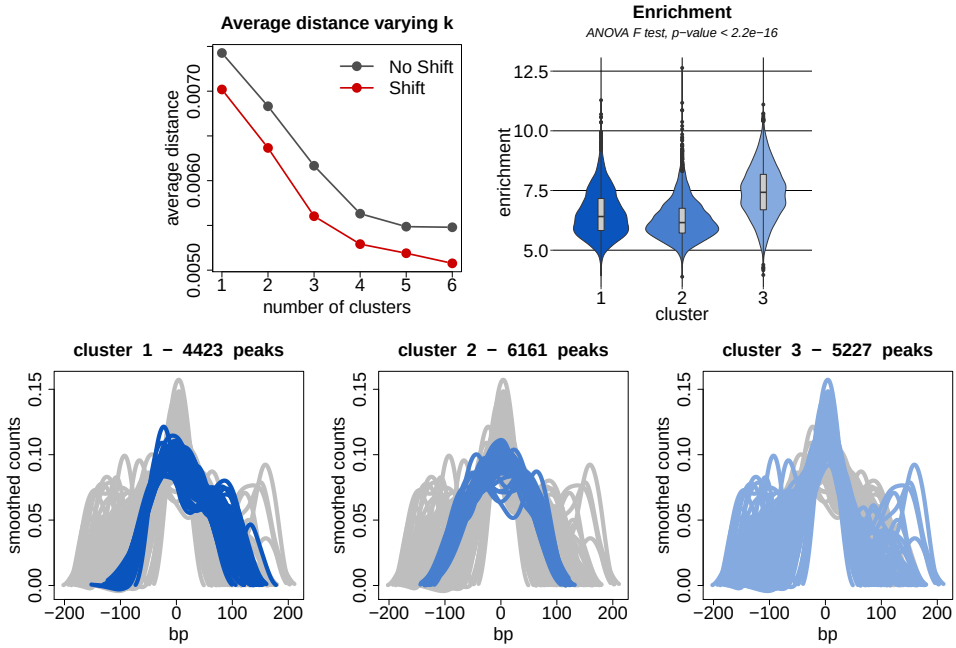
Moreover, in Figure S1.1 we present the results of the classification of peaks obtained setting the fragments length equal to the reads length during the piling up procedure (d of Equation (1.2.1) is set to 0) noticing that a wrong estimation of the parameter can cause the definition of too irregular peaks. We don't show the extended biological results since we notice that there is no connection between the classification obtained here and the biological inspections we introduced.

Finally, we applied the functional algorithm to a similar dataset, where Myc is overexpressed (MycER4h, Supplementary Figures S1.13), and we obtained a similar classification, where biological effects already observed are more pronounced (see Supplementary Figures S1.14 and S1.15). After Myc activation (MycER4h), the number of clusters remains the same, but a considerable number of overlapping peaks shift from the first cluster to the second, suggesting that the increased concentration of the TF can affect the shape of the peaks.

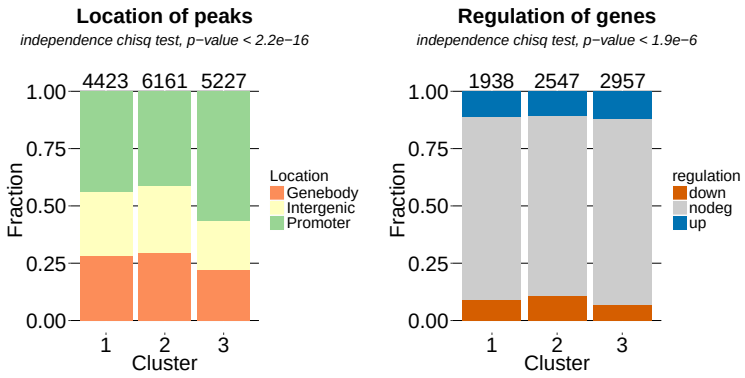
To conclude, the functional classification on a different TF (p53) in irradiated murine splenic B-cells identifies 2 clusters (Supplementary Figure 1.16) characterized by different levels of enrichment, presence of the binding motif and association to changes in gene expression (Supplementary Figures S1.17 and 1.18).



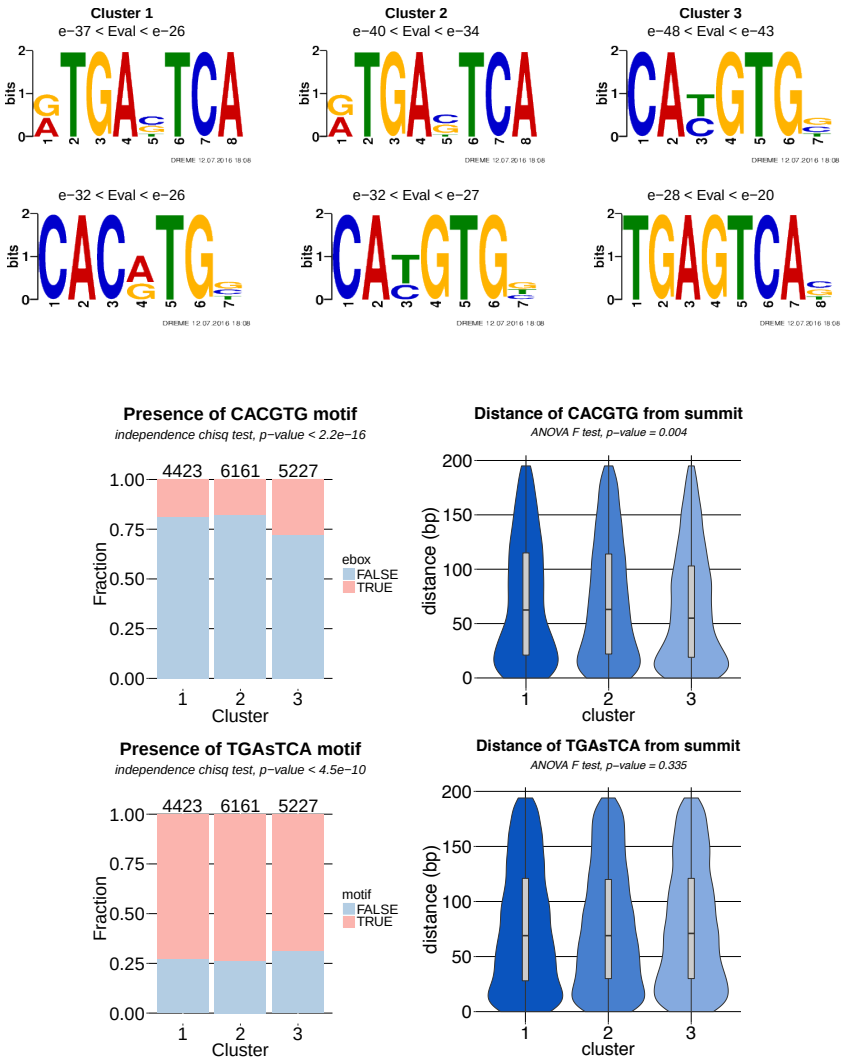
SUPPLEMENTARY FIGURE S1.1: Classification of MycER0h peaks with *FunChIP*. The coverage of the peaks is defined by the aligned reads, without the extension to the estimated fragment (equivalent to set $l = r$ in Equation (1.2.1)). Top left panel: distance within clusters, both with (red line) and without (black line) the alignment of peaks: in this case, the optimum number of clusters is $k = 5$ and the alignment of the peaks introduces a large shift between the two lines. Top right panel: the enrichment of the clusters are different, as the ANOVA F test has a p-value $< 10^{-16}$. Performing pairwise z tests and correcting p-values with the Bonferroni method, we conclude that pairs of enrichments are significantly different, except for the first and the second clusters (corrected p-value 0.188). The fourth and fifth clusters also show a weak significance for differences (corrected p-value 0.0324), while for all the other couples the p-values are $< 10^{-4}$. Second and third line panels: examples of peaks in the five clusters, centered around their summits and aligned with the estimated shift coefficient. Clusters differ on the width, magnitude and general shape of peaks.



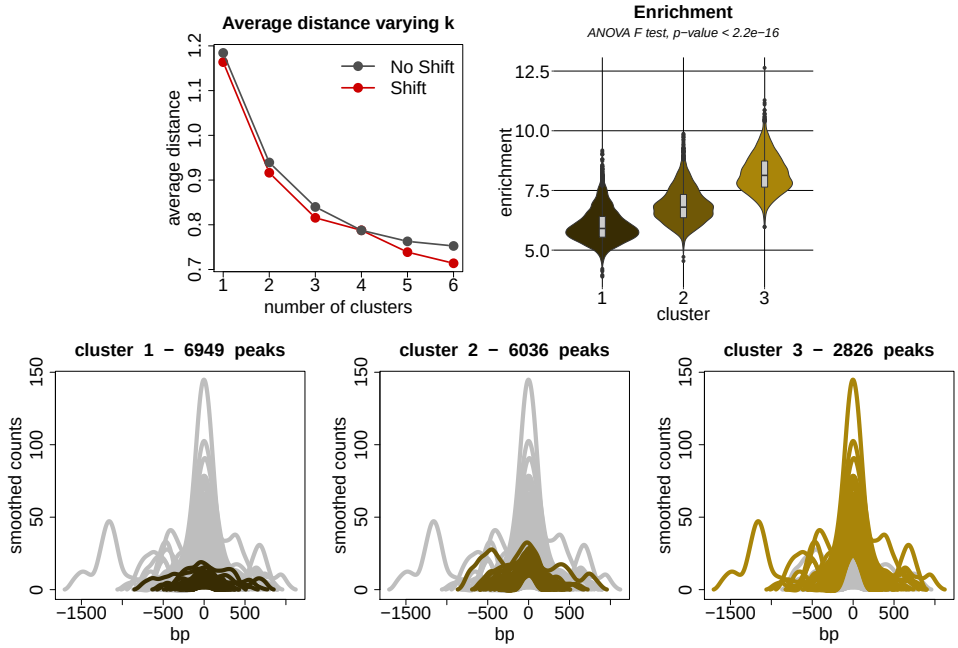
Supplementary Figure S1.2: Classification of MycER0h peaks with *FunChIP*. The coverage of the peaks is defined by piling up the fragments obtained by the extension of the reads collected from the ChIP-Seq experiment, with d estimated as in Equation (1.1). Peaks are then scaled to have the same width and area. Top left panel: distance within clusters, both with (red line) and without (black line) the alignment of peaks; in this case, the optimum number of clusters is $k = 3$ and the contribution of the alignment is significant. Top right panel: of the peaks in the three clusters is significantly different, as confirmed by the ANOVA F test; considering the pair-wise two-sided z-test for the differences in the averages and correcting the p-values with Bonferroni correction, all p-values are below 10^{-16}. Bottom panels: examples of peaks in the four clusters, centered around their summits and aligned with the estimated shift coefficient. Cluster 2 is composed by $\sim$ 39% of the peaks and contains the most regular unimodal peaks, while cluster 1 ($\sim$ 28% of peaks) contains regular data, but with an asymmetry towards the right side. Cluster 3 is composed by multimodal and irregular peaks.



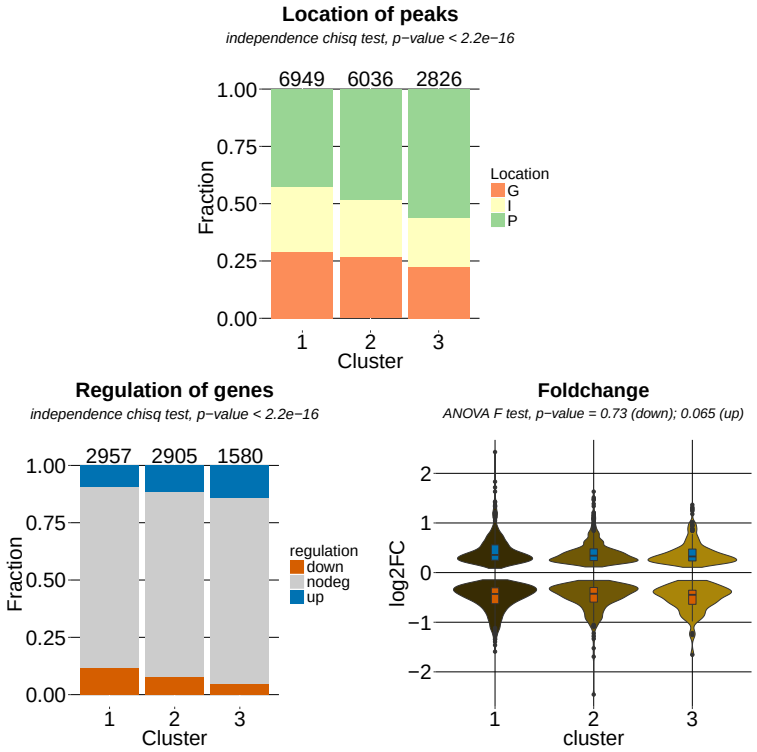
Supplementary Figure S1.3: Genomic location of peaks classified as described in Supplementary Figure S1.2. Both location and regulation of genes are different in the three clusters: χ^2 test for the differences in location has a p-value $< 10^{-16}$ and for the regulation $< 10^{-5}$. Left panel: peaks in cluster 3 seem to be more localized on promoter regions. The pairwise tests for the promoter regions on the three clusters, with the multiple correction of the χ^2 p-values, confirm a strong evidence to distinguish cluster 3 from 1 and 2 (the tests to test whether there is a difference in the classification of promoter peaks have p-values $< 10^{-16}$), while there is only a weak difference for cluster 1 and 2 (p-value = 0.035) . Right panel: peaks in cluster 3 tend to be less associated with down-regulated genes upon Myc overexpression. Comparing the presence of up-regulated genes, we notice that there is no evidence for any difference in the three clusters: all the pairwise χ^2 tests show correct p-values higher than 0.6; regarding the down-regulated genes, instead, we detect no significant difference for cluster 1 and 2 (correct χ^2 test p-value = 0.249), while cluster 3 is different both from cluster 1 (p-value $< 10^{-3}$) and cluster 2 (p-value $< 10^{-7}$).



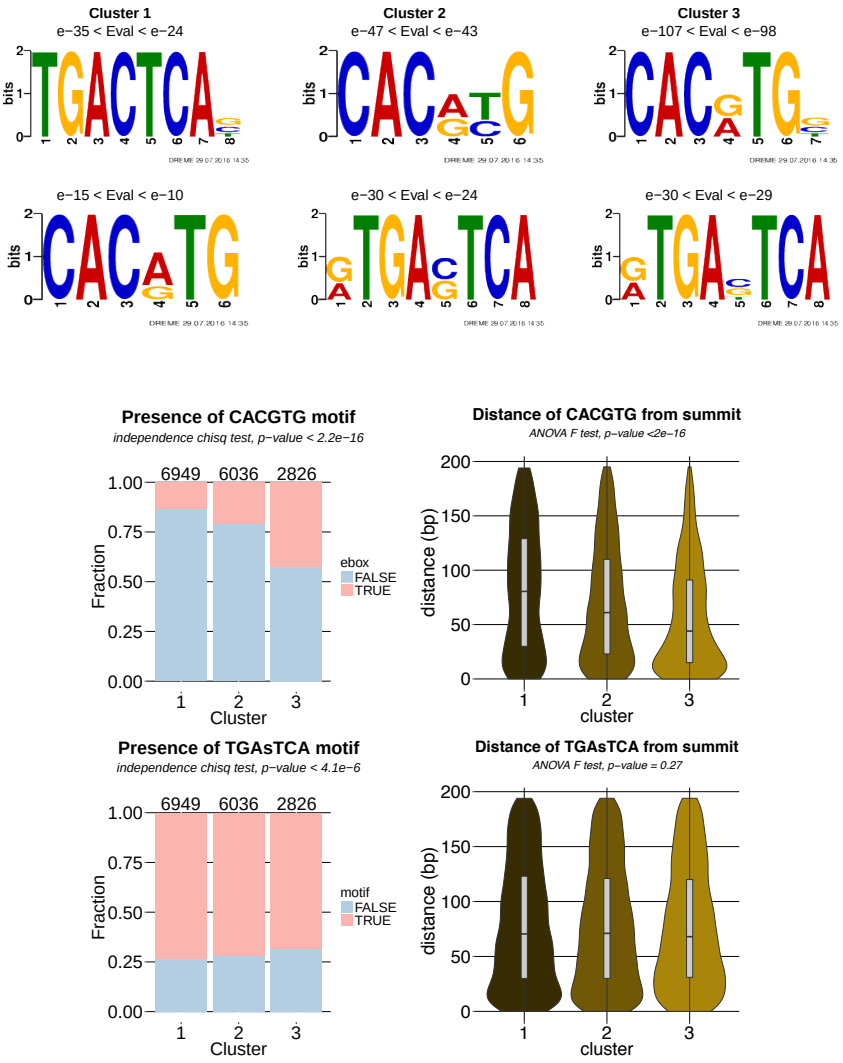
Supplementary Figure S1.4: Motif analysis of peaks classified as described in Supplementary Figure S1.2. Top panels: results of the motif discovery performed on the peaks of the two clusters. In cluster 1 and 2, the main motif detected is TGAsTCA (E-significance between 10^{-40} and 10^{-27}). The Myc binding motif, or E-box, instead, is detected with a lower significance. In cluster 3, the same two motifs are found, but with inverted order of significance (E-value $< 10^{-43}$ for E-box, and between 10^{-28} and 10^{-20} for the TGAGTCA motif). Middle panels: the presence of the E-box is different in the three clusters (χ^2 test p-value $< 10^{-16}$), and in cluster 3 it is more present and closer to the summit. The F test for the comparison of the distances in the three clusters shows a p-value of 0.004 and the third cluster is the responsible for this difference: both the z-test comparing the first and the second cluster and the one comparing the first and the third show small corrected p-values, respectively 0.0046 and 0.00045. Bottom panels: the TGAsTCA motif seems to be slightly more present in cluster 3, but equally distant from the summit of the peaks than in the other clusters.



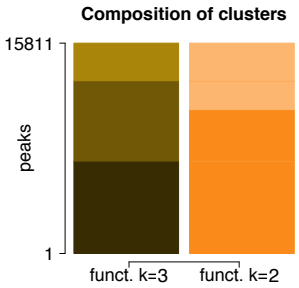
Supplementary Figure S1.5: Classification of MycER0h peaks with *FunChIP*. The coverage of the peaks is defined by piling up the fragments obtained by the extension of the reads collected from the ChIP-Seq experiment, with d estimated as in Equation (1.1). Top left panel: distance within clusters, both with (red line) and without (black line) the alignment of peaks; while in the main text we considered the case $k = 2$ with alignment, here we discuss the choice $k = 3$, with alignment. Top right panel: the enrichment of the peaks in the three clusters is significantly different, as confirmed by the ANOVA F-test; moreover, considering the pair-wise two-sided z-test for the differences in the averages and correcting the p-values with Bonferroni correction, all the p-values are below 10^{-16}. Bottom panels: examples of peaks belonging to the three clusters, centered around their summits and aligned with the estimated shift coefficient. Cluster 1 contains $\sim 44\%$ of peaks, which are small and irregular, cluster 2 contains $\sim 28\%$ of peaks, which are sharper and larger than those in Cluster 1, but less than those of Cluster 3 ($\sim 28\%$ of peaks).



Supplementary Figure S1.6: Genomic location of peaks classified as described in Supplementary Figure S1.5. Top panel: peaks in the clusters have different locations (χ^2 test p-value $< 10^{-16}$). The proportion of peaks on promoter regions is different in all the three clusters (all the pairwise χ^2 tests to test whether there is a difference in the classification of promoter peaks have p-value $< 10^{-9}$. Bottom left panel: the presence of regulated genes upon Myc overexpression is different in the three clusters (χ^2 test considering in the three clusters up, down and nodeg has a p-value $< 10^{-16}$). Moreover pairwise tests for the presence of down regulated genes show corrected p-value $< 10^{-4}$ for all the couples of clusters, regarding up-regulated genes, instead, there is strong evidence only for a difference in cluster 3 and 1, with a corrected p-value $< 10^{-5}$, while for the other two comparisons the evidence is weaker: for cluster 1 and 2 the p-value $= 0.024$ and for cluster 2 and 3 the p-value $= 0.051$. Bottom right panel: the logarithm of the fold change of up-regulated genes is on average slightly different in the three clusters, but it does not reach high significancy (ANOVA F test p-value 0.065).

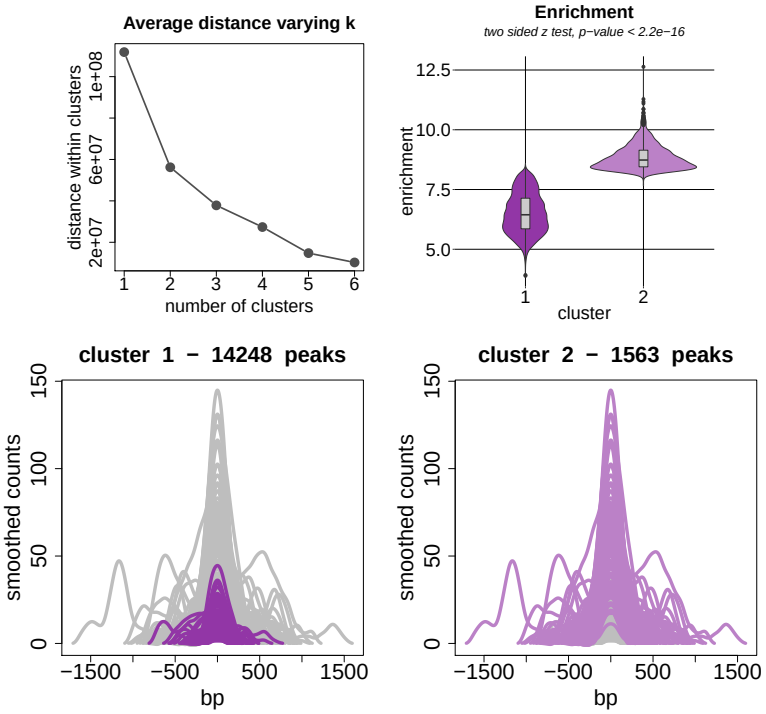


Supplementary Figure S1.7: Motif analysis of peaks classified as described in Supplementary Figure S1.5. Top panels: results of the motif discovery performed on the peaks of the three clusters. In cluster 1, the main motif detected is TGACTCA or TGAGTCA (E-significance between 10^{-35} and 10^{-24}). The Myc binding motif, or E-box, instead, is detected with a lower significance. In cluster 2 and 3, the same two motifs are found, but with inverted order of significance. We notice that the significance of the Myc binding motif is increasing with the increase of the regularity of peaks, from cluster 1 to cluster 3, while the TGACTCA has almost constant significance: E-value $\simeq 10^{-30}$. Middle panels: the E-box is present in different proportions in the three clusters , (χ^2 test p-value $< 10^{-16}$) and their average distance from the summit is different (ANOVA F test p-value $< 10^{-16}$). All the pairwise tests for these differences have a Bonferroni corrected p-value lower than 10^{-6}. Bottom panels: the TGAsTCA motif has slightly different presence in the three clusters, but no differences in the distances from summit are evident.

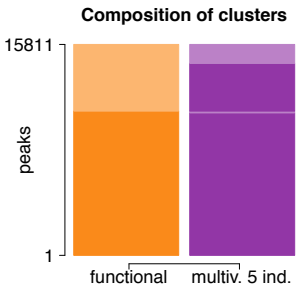


Functional Functional	Cluster 1	Cluster 2
Cluster 1	6949	0
Cluster 2	3865	2171
Cluster 3	0	2826

SUPPLEMENTARY FIGURE S1.8: Comparison between the classifications obtained with *FunChIP* on the MycER0h dataset with $k = 2$ (Figure 1.10) and $k = 3$ (Supplementary Figure S1.5), respectively. We notice that cluster 1 in the $k = 3$ case is entirely included in cluster 1 in the $k = 2$ case, as cluster 3 for $k = 3$ is included in cluster 2 for $k = 2$. The remaining cluster for $k = 3$ is equally split between the two cluster of $k = 2$, confirming that the extra cluster introduced for $k = 3$ mostly gather intermediate shapes between the two clusters obtained for $k = 2$.

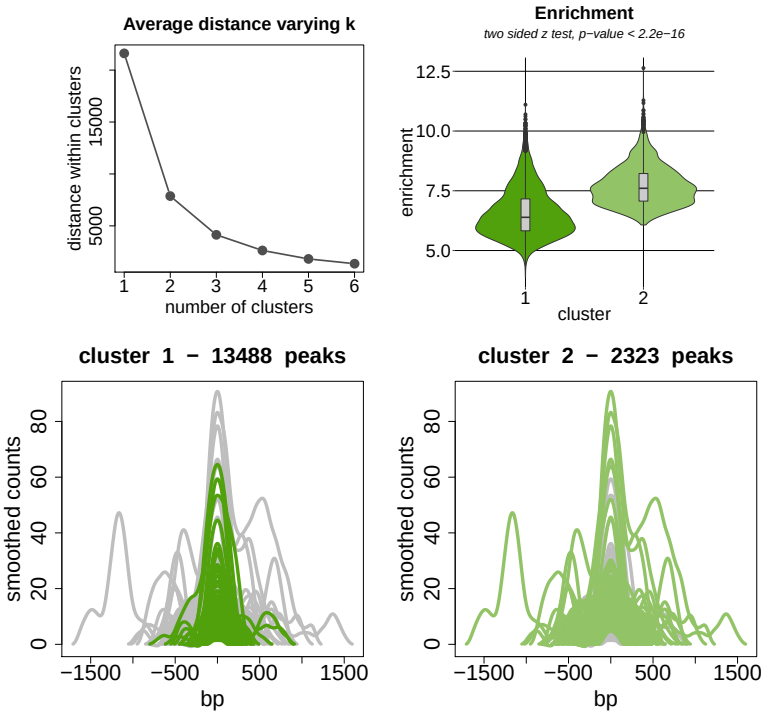


Supplementary Figure S1.9: Classification of MycER0h peaks with the multivariate algorithm presented in Cremona et al. (2015). Top left panel: distance within clusters; in this case, the optimum number of clusters is $k = 2$. Top right panel: distribution of the enrichment of the peaks in the two clusters, on average higher for cluster 2 (two-sided z-test p-value $< 10^{-16}$). Bottom panels: examples of peaks belonging to the two clusters, aligned around their summits. Cluster 1 contains $\sim 90\%$ of peaks, which are on average small and little enriched, while cluster 2 contains the higher and wider peaks (about $\sim 10\%$ of the total).

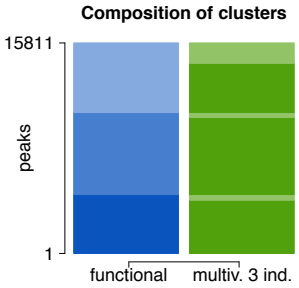


Multivariate / Functional	Cluster 1	Cluster 2
Cluster 1	10676	138
Cluster 2	3572	1425

Supplementary Figure S1.10: Comparison between the classification obtained with the multivariate method with 5 shape indices (Supplementary Figure S1.9) and that obtained with *FunChIP* (no scaling, reads extended in according to Equation (1.1), Figure 1.10). The second multivariate cluster, which is composed by wider and higher peaks, is almost included in the second cluster of *FunChIP*, composed by regular and well defined peaks, whereas the first multivariate cluster, composed by small peaks, includes all the elements of the first clusterof *FunChIP* (composed by small and irregular peaks), and part of the second cluster of *FunChIP*. We conclude that the functional classification focuses more on the regularity of the peaks instead of their size.

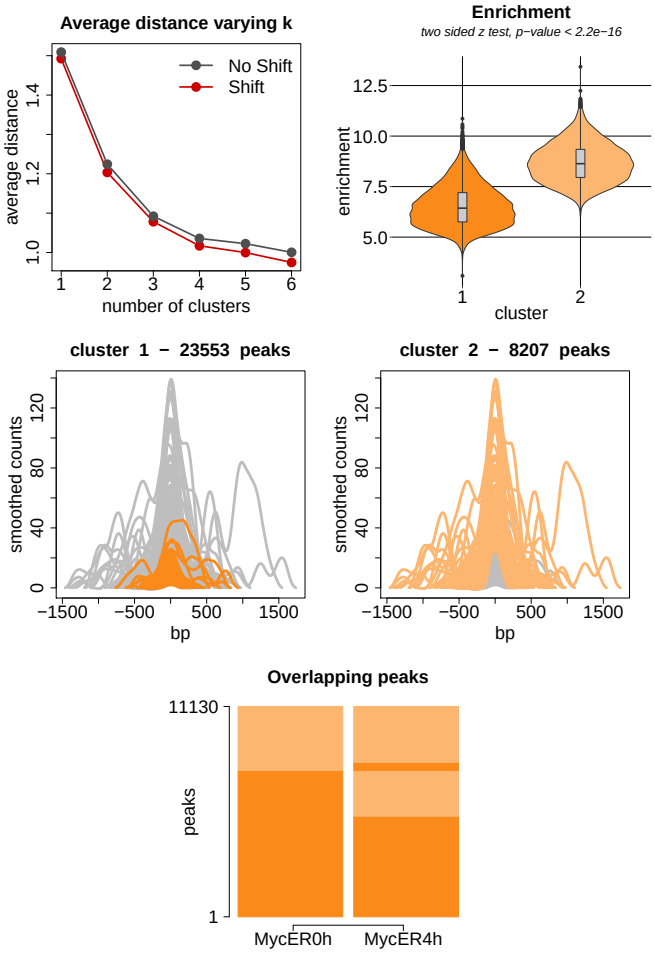


Supplementary Figure S1.11: Classification of MycER0h peaks with the multivariate algorithm presented in Cremona et al. (2015), removing 2 of the 5 indices (the area and the width of peaks). This classification should be more focused on the shape of peaks, rather than on their size. Top left panel: distance within clusters; in this case, the optimum number of clusters is $k = 2$. Top right panel: the distribution of the enrichment of the peaks in the two clusters, on average higher for cluster 2, is similar to what observed in Supplementary Figure S1.9, although the difference is less marked (two-sided z-test p-value $< 10^{-16}$). Bottom panels: examples of peaks belonging to the two clusters, aligned around their summits. Cluster 1 contains $\sim 85\%$ of peaks, which are now small and regular, while cluster 2 contains the higher and less regular peaks (about $\sim 15\%$ of the total).

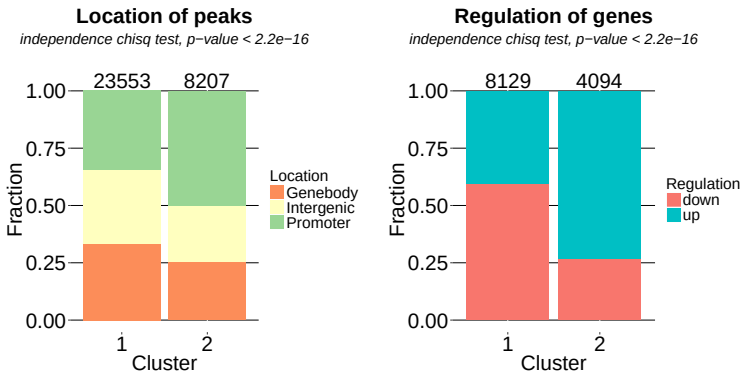


Multivariate / Functional	Cluster 1	Cluster 2
Cluster 1	4000	423
Cluster 2	5793	368
Cluster 3	3695	1532

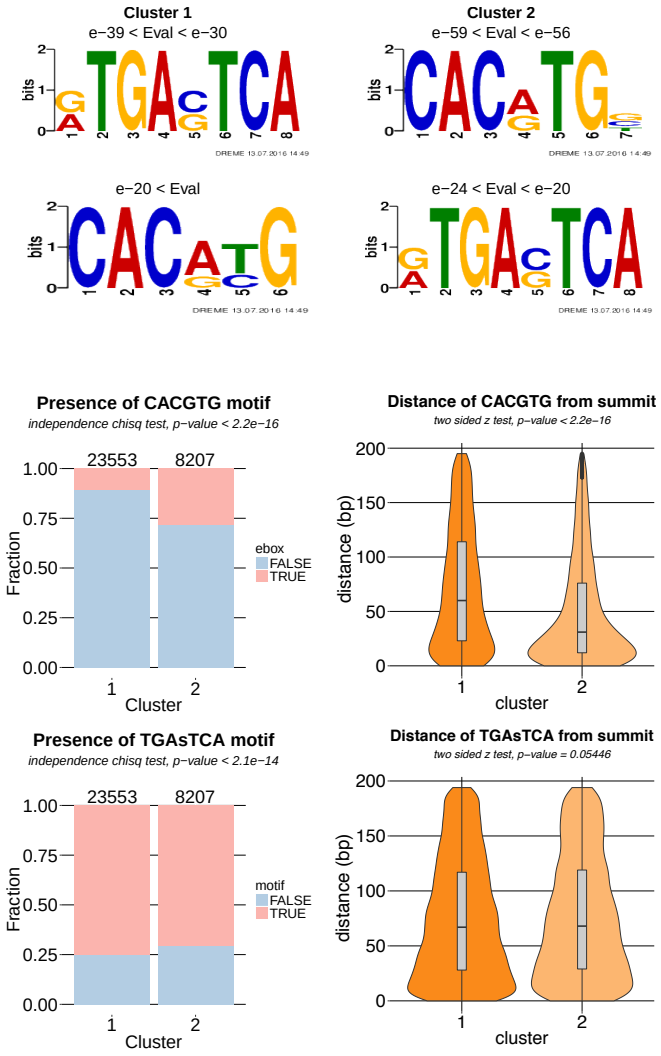
Supplementary Figure S1.12: Comparison between the classification obtained with the multivariate method with 3 shape indices (Supplementary Figure S1.11) and that obtained with *FunChIP* (scaling of peaks, reads extended in according to Equation (1.1), Supplementary Figure S1.2). The two classifications should be less sensitive to the magnitude of peaks. The second multivariate cluster, composed by large and irregular peaks is almost included in the third cluster of *FunChIP*, composed by multimodal and irregular peaks, whereas the first multivariate cluster, composed by small and regular peaks, is scattered in the three clusters *FunChIP*. We conclude that the multivariate classification, based on a representation of the peak with only 3 parameters does not capture the fine details of the shape of peaks, like the small asymmetry identified in cluster 1 with *FunChIP* (Supplementary Figure S1.2).



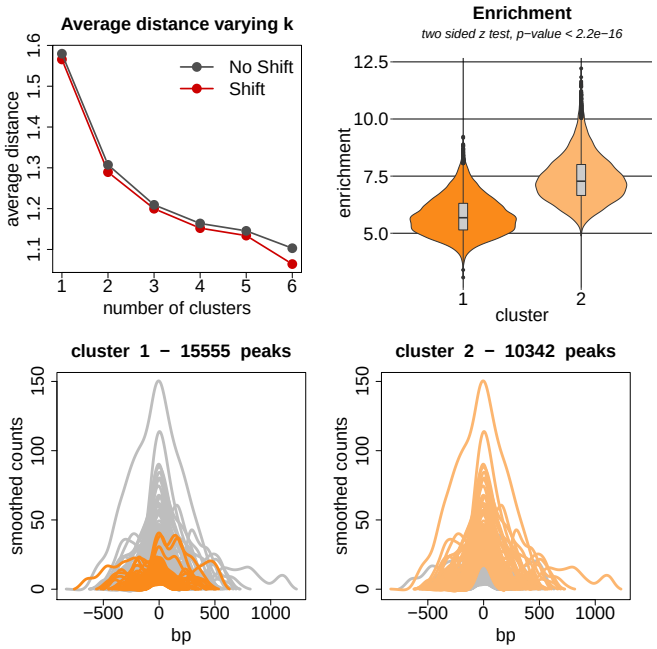
Supplementary Figure S1.13: Classification of MycER4h peaks with *FunChIP*. The coverage of the peaks is defined by piling up the fragments obtained by the extension of the reads collected from the ChIP-Seq experiment, with d estimated as in Equation (1.1). Top left panel: distance within clusters, both with (red line) and without (black line) the alignment of peaks; as in the main text Figure, the contribution of the alignment is modest, but it is still present. As for the main analysis we focus on $k = 2$, to compare the results with the MycER0h dataset, even if also the classification with $k = 3$ could be an interesting analysis. Top right panel: the enrichment of the peaks in the two clusters is on average higher for cluster 2 (two-sided z-test p-value $< 10^{-16}$). Middle panels: examples of peaks belonging to the two clusters, aligned around their summits. Cluster 1 contains $\sim$ 74% of peaks, which are small and narrow, while cluster 2 contains the higher and wider peaks. Bottom panel: analysis of common regions in the MycER0h and MycER4h dataset. The 15811 regions of peaks of MycER0h and the 31760 regions of peaks of MycER4h are analyzed together to select the common regions: the ones overlapping at least for one base pair of the genome. We isolate 11130 regions and here we plot the cluster they belong to. We see how some of the small and irregular peaks of MycER0h (Cluster 1) increasing the expression level of Myc (MycER4h) become part of the set of well defined and sharp peaks (Cluster 2). Cluster 2 of MycER0h, instead, becomes almost completely part of the Cluster 2 of MycER4h.



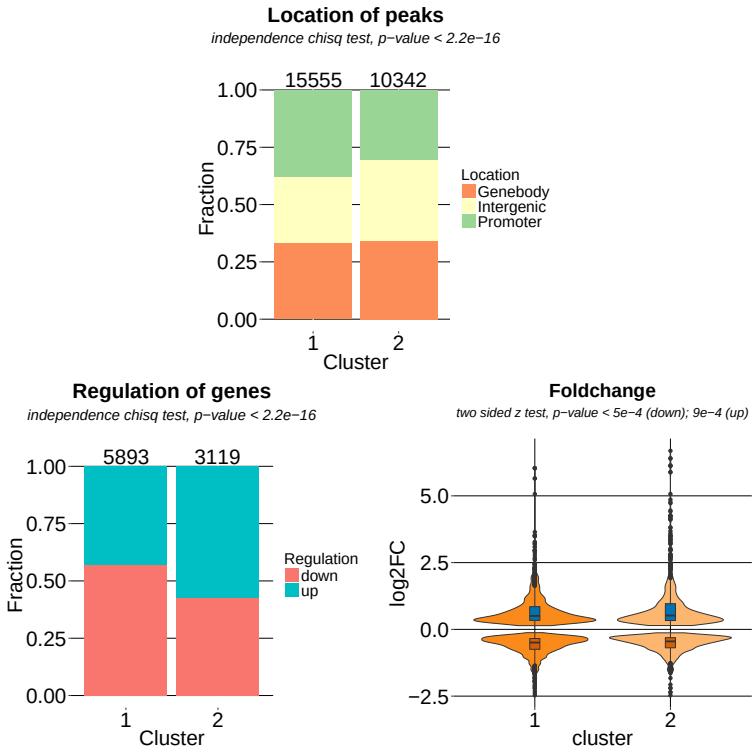
Supplementary Figure S1.14: Genomic location of peaks classified as described in Supplementary Figure S1.13. Left panel: peaks have different locations in the two clusters (χ^2 test p-value $< 10^{-16}$) and in particular, in cluster 2 peaks are more localized on promoter regions (χ^2 test to test whether there is a difference in the classification of promoter peaks has a p-value $< 10^{-16}$). Right panel: there is a difference in the regulation of peaks in the two clusters (χ^2 test p-value $< 10^{-16}$); peaks in cluster 1 tend to be more associated with up-regulated genes upon Myc overexpression.



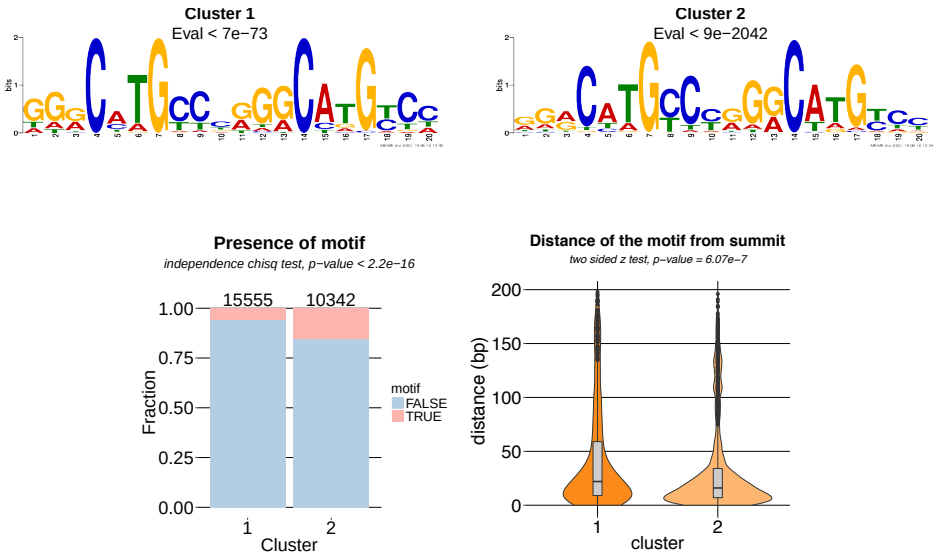
Supplementary Figure S1.15: Motif analysis of peaks classified as described in Supplementary Figure S1.13. Top panels: results of the motif discovery performed on the peaks of the two clusters. In the first cluster, the main motif detected is TGAsTCA (E-significance between 10^{-39} and 10^{-30}). The Myc binding motif, or E-box, instead, is detected with a lower significance. In cluster 2, the same two motifs are found, but with inverted order of significance (E-value $< 10^{-56}$ for E-box, and between 10^{-24} and 10^{-20} for the TGAGTCA motif). Middle panels: the E-box is overall less present than in the peaks of MycER0h, yet significantly more present in cluster 2, (χ^2 test p-value $< 10^{-16}$) and closer to the summit (z test p-value $< 10^{-16}$) than in cluster 1. Bottom panels: the TGAsTCA motif is slightly more present in cluster 2, but equally distant from the summit of the peaks than in cluster 1.



Supplementary Figure S1.16: Classification of p53ER peaks with *FunChIP*. The coverage of the peaks is defined by piling up the fragments obtained by the extension of the reads collected from the ChIP-Seq experiment, with d estimated as in Equation (1.1). Top left panel: distance within clusters: in this case, the optimum number of clusters is $k = 2$ and the contribution of the alignment is very small. Top right panel: the enrichment of the peaks in the two clusters is on average higher for cluster 2 (two-sided z-test p-value $< 10^{-16}$). Bottom panels: examples of peaks belonging to the two clusters, aligned around their summits. Cluster 1 contains $\sim 60\%$ of peaks, which are small and irregular, while cluster 2 contains the higher and well defined peaks.



Supplementary Figure S1.17: Genomic location of peaks classified as described in Supplementary Figure S1.16. Top panel: peaks of different clusters are differently localized (χ^2 test p-value $< 10^{-16}$), in particular there is strong evidence to confirm that there is difference for the two clusters in the location on intergenic and promoter regions (both χ^2 to test whether there is a difference in the classification of intergenic and promoter peaks have p-values $< 10^{-16}$), while there is no difference in the proportion of peaks localized on genebody regions (χ^2 p-value $= 0.104$) . Bottom left panel: there is difference in the association of peaks of the two clusters to up and down regulated genes with respect to non irradiated cells, where p53 is expressed at much lower levels (χ^2 test p-value $< 10^{-16}$). Cluster 2 is more associated to up-regulated genes with respect to cluster 1. Bottom right panel: the fold changes of up- and down-regulated genes are different for genes having peak of cluster 2 on their promoters (both z test p-values $\sim 10^{-4}$).



Supplementary Figure S1.18: Motif analysis of peaks classified as described in Supplementary Figure S1.16. Top panels: results of the motif discovery performed on the peaks of the two clusters. Both cluster display the p53 typical motif, but the significance is higher for cluster 2 (E-value $\sim 10^{-73}$ for cluster 1 and $< 10^{-100}$ for cluster 2). Bottom panels: the p53 binding motif is significantly more present in cluster 2 (χ^2 test p-value $< 10^{-16}$) and closer to the summit (z test p-value $< 10^{-6}$) than in cluster 1.

2

HETEROGENETY OF COGNITIVE DECLINE IN DEMENTIA: TAKING INTO ACCOUNT VARIABLE TIME-ZERO SEVERITY

2.1 INTRODUCTION

Long-term models of the progression of dementia are critically important for understanding prognosis, disease etiology, patient heterogeneity and the optimal design of trials. However, it is challenging to study cognitive decline due to short-term infrequent follow-up of individuals in existing studies and heterogeneity of decline. Additionally, individuals are recruited to studies at different ages, with different underlying diseases and at different times relative to disease onset. To date, two main methods have been used to develop models of cognitive decline: clustering approaches and disease progression score approaches.

An advantage of clustering approaches is that they naturally deal with heterogeneity, including unmeasured factors. A state of the art clustering approach used to model dementia progression is the Latent Class Mixture Model (LCMM). For example, to study trajectories of cognitive decline Wilkosz et al. (2009) fit a LCMM of quadratic trajectories to longitudinal Mini-Mental State Examination (MMSE) scores, with time zero set to be the date of the first assessment. This revealed six trajectories of cognitive decline in Alzheimer's disease (AD), of which the fastest declining had a greater number of patients with psychotic symptoms. However, many of these trajectories appear to differ only in their intercept, i.e. an individuals score at first visit. In other words, different clusters can represent differences in either progression or in disease stage at first presentation.
More recently, Proust-Lima et al. (2015) have used a spline LCMM to model decline in MMSE, with date at 65th birthday used as time zero. These models are used to generate four clusters, whose relation to the risk of developing

dementia is demonstrated. This analysis also leads to some trajectories that differ mostly in MMSE score at age 65, i.e. there exists groups of individuals who progress similarly but at an older age. In other words, existing clustering approaches generate trajectory models that are strongly confounded by MMSE score at time zero, independentely from how it is defined.

Disease progression score methods, instead, take an other approach, essentially using the longitudinal data itself to estimate time since disease onset (Yang et al. (2011), Jedynak et al. (2012)). For example, they assume that cognitive decline can be divided into two linear sections, representing normal decline and accelerated decline due to dementia respectively. The change point at which they meet is learned from the data. However, the assumption of linearity before and after the change point is probably unrealistic and potentially bias the results.

Rather than use a change point model, other researchers have assumed sigmoidal (i.e. logistic) or exponential trajectories of cognitive decline, with individual offsets (a random effect) used to align each individual to the trajectory model (Yang et al. (2011), Jedynak et al. (2012)). These models have been used to study the temporal relationship of different biomarkers, leading to models that are only partially consistent with the theoretical model of Clifford et al. (2013). Neither of these approaches has been used to develop a clustering approach to help identify risk factors affecting rates of cognitive decline.

A method that combines the clustering approach with the identification of the initial point of the disease progression could theoretically solve the problem of confounding score at time zero. Gaffney and Smyth (2004), Sangalli et al. (2010) and Kiddle et al. (2010) have demonstrated the importance of considering alignment in clustering procedures: their simulations show that if misalignment is present, clustering alone incorrectly groups observations. However, these methods depend on densely sampled regular timepoints and none have been designed to work with short-term follow-up relative to long-term progression.

In this study we combine the benefits of clustering and disease progression score approaches to study long-term cognitive decline in datasets with infrequent and irregular time points.
For example, cohort studies such as Alzheimer's Disease Neuroimaging Initiative (ADNI) of Mueller et al. (2005) and clinical trials such as those in the Coalition Against Major Diseases (CAMD) of Neville et al. (2015) have relatively short-term follow-up over months or a few years, while the course of cognitive decline in individual patients can take a decade or more (Wilkosz

et al. (2009)). We apply a novel method, called Temporal Clustering, which combines the benefits of clustering and an inferred *time zero* to improve the analysis of longitudinal heterogeneity. Further, we compare the ability of Temporal Clustering and LCMMs to generate clusters that are related to AD risk factors. To our knowledge this is the second application of such a method to study disease progression (Huopaniemi et al. (2014)); but it is the first attempt to use this type of approach to study cognitive decline. All analyses are done with open source code, available on to the GitHub repository of Kiddle (2016), to allow our work to be verified and extended by others.

This chapter is organized as follows. In Section 2.2 we present the classification method that, beside classifying patients, identifies the individual offsets. In Section 2.3 we describe the composition of the global dataset we will examine. In Section 2.4 we present some simulation studies and in Section 2.5 we illustrate the application of the classification method to the real dataset, with the biological inspection of the results. Finally in Section 2.6 some global considerations on the results and on further developments are presented.

2.2 METHODS

The classification method we propose in this work is the Temporal Clustering method. For this method a parametrized trajectory curve ($\phi(t;\theta)$) for the MMSE is supplied, where t is time and θ is a vector of trajectory parameters.

For this application we first used a three parameter sigmoidal trajectory, based on the assumptions of Clifford et al. (2013):

$$\phi(t;\theta) := \frac{\theta_1}{1+\exp(-\theta_2 t)} + \theta_3;$$

then we found that a two parameter exponential decline curve fits equally well

$$\phi(t;\theta) := \theta_1 - \exp(\theta_2 t),$$

and we therefore use this for inference since it requires fewer parameters. Here, θ_1 represents the maximum MMSE of the trajectory model and θ_2 is an exponential decline rate.

Given a dataset containing information on individuals $i \in \{1, 2, \ldots, N\}$, the set of time points at which individual i has been observed is denoted $\boldsymbol{\tau}^i := (\tau^i_1, \ldots, \tau^i_{\#TP(i)})$. The longitudinal univariate observations of the MMSE score for individual i at time t are given by $x_i(t)$, which is only

Algorithm 2: Temporal Clustering

Choose number of clusters K.

1. First fix $\hat{\theta}$ by estimating it across all individuals by minimizing with respect to θ the following equation

$$\text{loss}(\theta) = \sum_{i=1}^{N} \|x_i(\boldsymbol{\tau}^i) - \phi(\boldsymbol{\tau}^i + \delta_i; \theta)\|_2^2 \tag{2.1}$$

2. Estimate the optimal offset for each individual i as:

$$\hat{\delta}_i = \arg\min_{\delta} \|x_i(\boldsymbol{\tau}^i) - \phi(\boldsymbol{\tau}^i + \delta; \hat{\theta})\|_2^2 \tag{2.2}$$

3. Randomly assign each individual to a cluster,
i.e. sample $c(i)$ for all i randomly with replacement from $\{1, \ldots, K\}$.

4. Repeat the following steps up to convergence, defined as no change in $c(i)$.

   a. For each cluster k, minimise Equation (2.1) with respect to $\theta_{-1}^k = \theta_2^k$ (θ_1 is constant and equal for all the clusters) using only data from individuals for whom $c(i) = k$:

$$\text{loss}(\theta_2^k) = \sum_{i:c(i)=k} \|x_i(\boldsymbol{\tau}^i) - \phi(\boldsymbol{\tau}^i + \delta_i; \theta_2^k)\|_2^2$$

   b. For each cluster k and individual i, use θ^k to estimate the individual offsets $\{\hat{\delta}_i^k(\theta^k), k = 1 : \ldots, K\}$ and the the local_loss associated to each cluster k:

$$\text{local\_loss}(i, \hat{\delta}_i^k, \theta^k) = \|x_i(\boldsymbol{\tau}^i) - \phi(\boldsymbol{\tau}^i + \hat{\delta}_i^k; \hat{\theta}^k)\|_2^2. \tag{2.3}$$

   Then, reassign each individual i to the cluster k_i' which results in the smallest `local_loss`:

$$c(i) \leftarrow k_i' = \arg\min_{k=1:K} \text{local\_loss}(i, \hat{\delta}_i^k, \theta^k)$$

   .

observed for $t \in \boldsymbol{\tau}^i$. The $\phi(t;\theta)$ parametrized trajectory, along with an individual's offset δ_i, is believed to explain the data, i.e. $x_i(t) \approx \phi(t+\delta_i;\theta)$. And then the estimated Disease Time $\widehat{DT}$ at time t for the individual i is: $\widehat{DT}(i,t) := t + \delta_i$.

The Temporal Clustering model estimates the trajectory parameters θ^k for each cluster k, and the offsets δ_i for each individual i. These offsets are used to shift time points to better align individuals to cluster trajectories, i.e. $\phi(t+\delta_i;\theta^k)$ is the expected MMSE score at time point t for individual i in cluster k. The introduction of this shift coefficient allows to estimate a common starting point of the cognitive decline of patients, without fixing it as the first clinical exam or as a specific year of age.

Using the exponential decline trajectory, the offset $\hat{\delta}_i$ for an individual i is an estimate of the time between first MMSE assessment and the time at which their MMSE score reached one MMSE point below the maximum MMSE of the model, i.e. an MMSE of $\theta_1 - 1$.

A simplifying assumption of Temporal Clustering is that the baseline parameter θ_1 takes the same values across all clusters, this was found to be necessary to get good performance and provide identifiability given short follow-up, as we present in Section 2.4.

The Temporal Clustering algorithm (presented in Algorithm 2) is based on K-means, a commonly used clustering algorithm. K-means finds clusters by initially allocating all individuals to K clusters at random, and then iterating two steps until the model converges. In K-means the step a. involves calculating a mean point over all cluster members, and step b. involves re-allocating individuals to the cluster whose mean they are closest to.

The introduction of the estimation of the individual offset $\hat{\delta}_i$ at each iteration of the K-mean algorithm allows to consider the registration problem simultaneously with clustering, as proposed by Sangalli et al. (2010). The difference between the K-mean alignment algorithm and Temporal Clustering is that rather than calculating cluster means in Step a., as the functional mean or medoid, Temporal Clustering infers trajectory parameters $\hat{\theta}^k$ for each cluster k, along with individual offsets $\hat{\delta}_i$.
This iterates with cluster re-assignment of Step b. until convergence.

In this study, for simplicity, we seek to split individuals into a groups of slower and relatively faster decliners (i.e. we fix the total number of cluster at $K = 2$). A discrimination score is then calculated to assess the relative quality of fit of each individual's data to their assigned versus unassigned

cluster. It consists of the absolute value of the difference of `local_loss` of Equation (2.3) for assigned cluster $c(i)$, and closest other cluster

$$\mathcal{D}_i = |\text{local\_loss}(i, \hat{\delta}_i^{c(i)}, \theta^{c(i)}) - \min_{k \neq c(i)} \text{local\_loss}(i, \hat{\delta}_i^{k}, \theta^{k})| \qquad (2.4)$$

2.3 DEFINITION OF A GLOBAL DATASET

We combine data including time series of MMSE from three prospective cohort studies

- Alzheimer's Disease Neuroimaging Initiative -ADNI- database (Mueller et al. (2005)) is launched in 2003 as a public-private partnership, led by Principal Investigator Michael W. Weiner, MD. The primary goal of ADNI is to test whether serial magnetic resonance imaging (MRI), positron emission tomography (PET), other biological markers, and clinical and neuropsychological assessment can be combined to measure the progression of mild cognitive impairment (MCI) and early Alzheimer's disease.
  For up-to-date information, see http://www.adni-info.org.

- Australian Imaging, Biomarkers and Lifestyle Flagship Study of Ageing -AIBL- (Ellis et al. (2009)) is a study launched in 2006 to discover which biomarkers, cognitive characteristics, health and lifestyle factors determine subsequent development of symptomatic Alzheimer's Disease.
  For up-to-date information, see http://www.aibl.csiro.au/adni/index.html.

- the Coalition Against Major Diseases -CAMD- database of Neville et al. (2015) is launched in 2008 and provides data only for the placebo arm of AD trials. It provides no further information about the trials or individuals diagnosis at first visit. To ensure sufficient follow-up, we only used data from the two longest running clinic trials in the CAMD database (C-1013 and C-1014). The coalition is formed by the Critical Path Institute in collaboration with the Engelberg Center for Health Care Reform at the Brookings Institution and brings together patient groups, biopharmaceutical companies, and scientists from academia, the U.S. Food and Drug Administration (FDA), the European Medicines Agency (EMA), the National Institute of Neurological Disorders and Stroke (NINDS), and the National Institute on

Aging (NIA).
For up-to-date information, see http://c-path.org/programs/camd/.

These three studies are combined to make a large dataset, all with short follow-ups, but with a lot of variability in baseline cognitive ability. They could be combined as MMSE was recorded consistently across all studies. Individuals are classified as having Normal Cognition (NC), Mild Cognitive Impairment (MCI) or Alzheimer's disease (AD). In CAMD diagnoses are not given, but they can be inferred based on MMSE in the way suggested in Tombaugh and McIntyre (1992). For all datasets we only extracted data for individuals with more than one time point containing non-missing data.

For ADNI and AIBL, genomic DNA was extracted from whole blood with *APOE* genotyped using either TaqMan probes for Single Nucleotide Polymorphisms (rs429358, rs7412) or the *Hha1* restriction enzyme, and assessed using Polymerase Chain Reaction (Frisoni et al. (2007), Gupta et al. (2015)). Levels of total Tau in cerebrospinal fluid from ADNI participants are also measured using the xMAP Luminex platform.

2.4 A SIMULATION STUDY

We performed simulations based on the combined cohort structure of ADNI and AIBL datasets, i.e. with the same number of individuals, similar time points and with longitudinal MMSE data resembling the real data.

Two types of simulation study are performed, a disease time estimation simulation (equivalent to $K = 1$) and a Temporal Clustering simulation (i.e. $K = 2$). The accuracy of clustering was assessed using the Adjusted Rand Index (ARI) of Rand (1971), which takes values from zero (i.e. no better than chance) to one (i.e. perfect clustering). ARI was calculated using the R package `mclust` (Fraley and Raftery (2002)).

First, vectors of true δ are uniformly sampled from an interval chosen to generate visually plausible data. The interval chosen depended on the value of true θ_2 and is summarized in Table 2.1.

To generate time points, simulated individuals are randomly assigned the time points of a real individual $\boldsymbol{\tau}$ in such a way that the last time point would have MMSE > 0 (before error was added).

Data are, then, generated based on our model, i.e.

$$x_i(\boldsymbol{\tau}^i) = \phi(\boldsymbol{\tau}^i + \delta_i; \theta) + \epsilon_{i\boldsymbol{\tau}^i}, \tag{2.5}$$

where $\epsilon_{i\boldsymbol{\tau}^i} \sim N(0, 1.5)$ i.i.d..

$\theta_2 \times 10^4$	Minimum $\delta \times 10^3$	Maximum $\delta \times 10^3$
1	10	30
3	0	10
5	0	6
6	0	5
7	0	4
8	0	3.5
9	-1	3.5
10	-1.5	3

TABLE 2.1: Table of delta ranges used to simulate data. Range used depended on θ_2.

For all the simulations true θ_1 is set to 29. For both the disease time estimation and Temporal Clustering simulations a range of five θ_2 is used to test the methods sensitivity to this parameter. For each θ_2 100 simulations are performed. For the disease time estimation we use $\theta_2 \times 10^4 = 1, 3, 5, 7$ and 9. For the Temporal Clustering simulation we use for the two clusters $\theta_2^1 \times 10^4 = 5, 3, 5, 6, 8$, and $\theta_2^1 \times 10^4 = 5, 1, 1, 10, 10$.

The simulation with $\theta_2^1 = \theta_2^2 = 5 \times 10^4$ is used to test what would have happened if there was only one cluster, but the user attempted to find two. This simulation is only used to study trajectory parameter estimation, and is not used in the clustering accuracy calculations as the cluster labels would have been meaningless.

To study the impact of various biases we investigate three different data transformations that led to progressively more realistic and biased data in the disease time estimation simulation . The first transformation simulates a floor effect by rounding the few cases where MMSE < 0 , due to the addition of noise, up to zero. The second also rounds each $x_i(t)$ to its nearest integer, as MMSE can only take integer values. The third additionally imposes a ceiling of MMSE = 30, the maximum score from the real MMSE test, i.e. all $x_i > 30$ are set to 30. For the Temporal Clustering simulation we only generate simulated data with all three of these transformations applied.

In a simulation the disease time estimation approach works best with a δ range of $\pm 20,000$ or $\pm 50,000$ days (Supplementary Figures S2.1 - S2.2), especially when the true rate parameter is low (e.g. $\theta_1 = 1 \times 10^{-4}$; Supplementary Figures S2.1A and S2.2A). This result is reasonable given the slow decline of the disease; MMSE can take decades to decrease significantly.

In a two-cluster simulation the fixed baseline is better than the separate baseline version of Temporal Clustering at avoiding unrealistic θ_1 estimates, i.e. $\theta_1 > 30$ (Supplementary Figure S2.3). In cases where accurate rate

estimation is the priority, it could make sense to use the separate baseline version, but the fixed baseline version is preferred here as it leads to trajectory parameters being interpretable, i.e. within realistic bounds.
After using simulated data to choose default values for Temporal Clustering parameters, we then used it to assess clustering accuracy, i.e. to assess whether Temporal Clustering could accurately distinguish between a slower and faster group of cognitive decliners. With no filter applied, the clustering result was only slightly better than would be expected by chance (ARI 0.12; Figure 2.1).

Clustering accuracy increased as individuals are filtered out on the basis of the discrimination score of Equation (2.4), as presented in Figure 2.1. A tradeoff can clearly be seen where the more stringent the filter (i.e. the higher the threshold) the higher the clustering accuracy and the lower the number of individuals left after filtering. For this reason we select a discrimination score threshold of 2, which led to a median clustering accuracy (ARI) of 0.31 but retained over half the individuals (median 54%). A much more stringent threshold of 10 led to a clustering accuracy (ARI) of 0.80, but retained fewer than a fifth of individuals on average.

The most obvious characteristics of simulated individuals removed by a discrimination score filter of 2 are that they have higher MMSE scores at first visit and/or less than a year of follow-up. This makes the classification for these individuals hard to be performed: not sufficient informations on the cognitive decline of patients have been collected.

2.5 REAL CASE STUDIES

2.5.1 *Temporal Clustering to distinguish between faster and slower decliners*

We next seek to use Temporal Clustering ($K = 2$) to summarize cognitive decline in the combined cohort. Before the discrimination score filter is applied, the classification resulted in one slowly declining cluster containing 1,335 individuals and another which declined faster and contained 1,077 individuals. The estimated maximum MMSE ($\hat{\theta}_1$) of the model was 30. The trajectory of the faster declining cluster takes approximately 15 years to go from an MMSE of 29 to 0, which fits well with LCMM trajectories learned from AD patients by Wilkosz et al. (2009), whereas the more slowly declining trajectory is estimated to take approximately 60 years. An example of the clustering results is shown in Figure 2.2 where also the estimated trajectories of the fast and slow decliners are plotted.

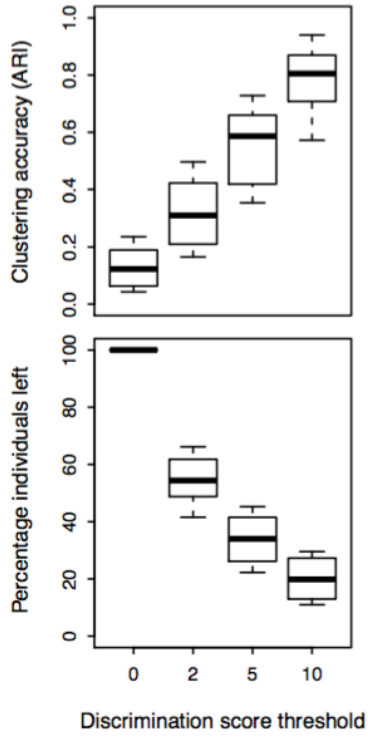


FIGURE 2.1: Boxplots of simulation study examining the effect of a discrimination score filter on the accuracy of Temporal Clustering cluster assignment, and number of individuals left after the filter is applied. Discrimination score is measure of relative goodness of fit of each individual to their assigned cluster. Clustering accuracy is measured in Adjusted Rand Index (ARI). Boxplots are over 100 simulations of four different choices of θ_2^1 and θ_2^2.

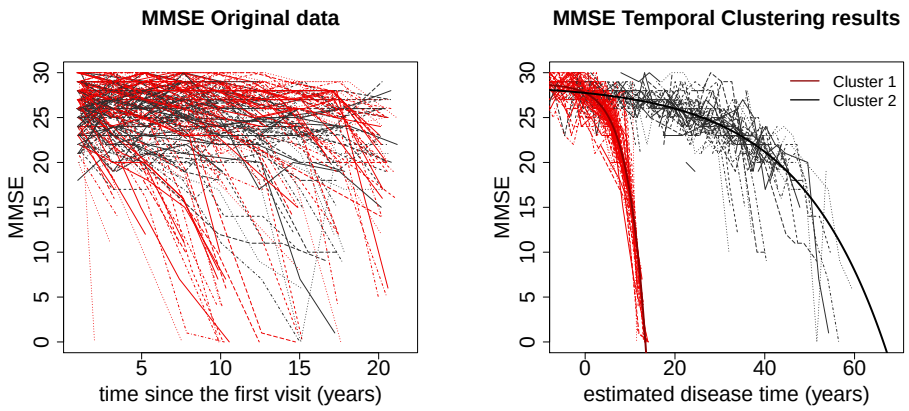


FIGURE 2.2: Left panel: plot of a subset of longitudinal MMSE measurements of a random subset of 400 data. Colors correspond to the final classification via the Temporal Clustering algorithm. Right panel: MMSE measurements aligned with the estimated individual offset. Lines corresponds to the parametrized trajectory estimated for the two clusters..

As our simulation study suggests, we filter out individuals for whom we do not have sufficient confidence in cluster assignment (i.e. those with a discrimination score < 2). This results in the removal of 31% of individuals, leaving 969 members (73%) of the slowly declining cluster and 688 members (64%) of the faster declining cluster. As in the simulation study, the filter removes a greater number of individuals with higher MMSE at first visit or individuals with very short follow up, presumably as the differences between clusters at that stage are more subtle.

We examined diagnosis at last visit just for individuals in ADNI1 or AIBL who remained after the filter, as these were the only cohorts with diagnostic information provided (instead of inferred). Approximately half of these individuals in the slow declining cluster had AD at last visit (99/191), in comparison to $\sim$ 80% (236/301) in the faster declining cluster.

2.5.2 *Association between AD risk factors and AD-like cognitive decline*

We next seek to use the filtered Temporal Clustering results to identify risk factors distinguishing the two clusters. Higher MMSE at first visit is associated with membership of the faster declining cluster (log odds ratio (LOD) of Edwards (1963) for 10 point increase $= 0.54$, p-value = 1.7×10^{-3}). Focusing on the subset of individuals with *APOE* data, we show a positive and

dose dependent association between the number of *APOE* $\epsilon 4$ alleles and membership of the faster declining cluster (1 allele LOD = 0.67, p-value = 1.3×10^{-3}; 2 alleles LOD = 0.81, p-value = 7.8×10^{-3}). Finally, we find an association between the fast declining cluster and the level of cerebrospinal fluid tau at first visit in ADNI (LOD for 100 pg/ml increase = 0.84, p-value = 0.014). Results are consistent when no filter is used and the global dataset is considered, see Table 2.2.

Moreover, we wish to compare the results of Temporal Clustering and LCMM. Specifically, we analyze which approach produced clusters with higher associations to known AD risk factors (*APOE* and Tau). To make the comparison more straightforward, this is performed on Temporal Clustering results before filtering, and on LCMM models without any co-variates. A synthesis of this comparison is shown in Table 2.2. To make it a fairer comparison for LCMM we used two different choices of *time zero* (first visit and 50th birthday) and both raw and normalized MMSE (normMMSE), as suggested in the original papers.

Because Temporal Clustering infers a *time zero*, we would expect by design its clusters to be less affected by MMSE at first visit than approaches like LCMM that do not. This is indeed the case, confounding of unfiltered Temporal Clustering clusters by MMSE at first visit is at least three-fold lower in absolute terms than that achieved by LCMM (LOD for a 10 point change = 0.5 for Temporal Clustering versus -1.6 or -10 for LCMM, Table 2.2). Indeed, MMSE at first visit is by far the most significant predictor of LCMM cluster membership in all cases. The biggest difference in the association of risk factors with unfiltered clusters was for *APOE*, especially the significance of the association of a single *APOE* $\epsilon 4$ allele with cluster membership (p-value = 1.1×10^{-5} for Temporal Clustering versus 0.62 or 0.013 for LCMM).

Overall LCMM cluster membership only has a clear relationship with AD risk factors when MMSE was normalized, which resulted in a large difference in MMSE at first visit between the clusters. In contrast Temporal Clustering cluster membership has a clear relationship to AD risk factors, and a lower difference in MMSE at first visit, with and without a discrimination score filter.

2.6 DISCUSSION

We have introduced a new method - Temporal Clustering - that can model cognitive decline by combining an estimated individual offset with clustering on that new time-scale. We show that this leads to clusters that are less

Method	Time zero	Variable	Risk factor	N	LOD	SE	p-value
TC	Inferred	MMSE	MMSE first visit	2399	0.5	0.14	4.6×10^{-4}
TC	Inferred	MMSE	1 APOE e4	1049	0.61	0.14	1.1×10^{-5}
TC	Inferred	MMSE	2 APOE e4	1049	0.84	0.23	3.6×10^{-4}
TC	Inferred	MMSE	CSF tau	407	0.77	0.24	0.0012
LCMM	First visit	MMSE	MMSE first visit	2399	-1.6	0.23	1.3×10^{-11}
LCMM	First visit	MMSE	1 APOE e4	1049	0.4	0.33	0.22
LCMM	First visit	MMSE	2 APOE e4	1049	0.64	0.41	0.12
LCMM	First visit	MMSE	CSF tau	407	0.38	0.4	0.35
LCMM	Age 50yrs	MMSE	MMSE first visit	2399	-1.7	0.25	3.5×10^{-12}
LCMM	Age 50yrs	MMSE	1 APOE e4	1049	0.17	0.34	0.62
LCMM	Age 50yrs	MMSE	2 APOE e4	1049	0.31	0.45	0.49
LCMM	Age 50yrs	MMSE	CSF tau	407	0.49	0.4	0.21
LCMM	First visit	normMMSE	MMSE first visit	2399	-10	0.72	5.0×10^{-48}
LCMM	First visit	normMMSE	1 APOE e4	1049	0.68	0.27	0.013
LCMM	First visit	normMMSE	2 APOE e4	1049	1.1	0.36	0.0017
LCMM	First visit	normMMSE	CSF tau	407	1.2	0.37	0.0015
LCMM	Age 50yrs	normMMSE	MMSE first visit	2399	-10	0.7	3.7×10^{-47}
LCMM	Age 50yrs	normMMSE	1 APOE e4	1049	0.62	0.27	0.022
LCMM	Age 50yrs	normMMSE	2 APOE e4	1049	1.1	0.35	0.0028
LCMM	Age 50yrs	normMMSE	CSF tau	407	1.1	0.36	0.0023

TABLE 2.2: Table summarising logistic regression analysis, comparing cluster membership to AD risk factors for Temporal Clustering (TC) and LCMM. Four different LCMM models have been run, combining one of two choice for *time zero* with the choice to use raw MMSE or normalized MMSE (normMMSE). Each line refers to a different logistic regression analysis to better cater for missing risk factor data, except for *APOE* for each clustering method, which were modelled together. Signs for LOD have been swapped when appropriate to allow appropriate comparisons, as signs depend on cluster labels which can be swapped arbitrarily. MMSE at first visit is coded in units of ten and cerebrospinal fluid tau is coded in units of 100 pg/ml.

influenced by MMSE at first visit, which we believe makes it easier to identify risk factors of cognitive decline. To this end we show a dose-dependent enrichment of *APOE* $\epsilon 4$ carriers in the faster declining (i.e. AD-like) cluster, a difference that is more significant than for clusters produced by LCMM approaches on this dataset.

There is some inconsistency in the literature about the relationship between *APOE* $\epsilon 4$ and the rate of cognitive decline. Some studies in non-demented individuals have found no relationship, e.g. Maria et al. (2002). However, the majority of studies have either found a modest relationship (e.g. Albrecht et al. (2015), Christensen et al. (2008), Gui et al. (2014), Carrasquillo et al. (2015)), or one that depends on other factors such as amyloid beta by Lim et al. (2015), alcohol by Downer et al. (2013) and body mass by Rajan et al. (2014). The inconsistency of these studies may be explained by cohort differences and/or the strong methodological challenges of the study of cognitive decline of Weuve et al. (2015).

In the field of cluster analysis determination of the optimal number of clusters is known to be tricky. For example, Bauer and Curran (2003) have argued that the optimal number of clusters in a model do not necessarily respond to the number of 'real' subgroups in an application. Instead, they argue that clusters can equally well be interpreted as having no meaning beyond being a convenient summary of non-gaussian distributions. Therefore in this study, for simplicity, we have generated models with just two clusters ($K = 2$). By comparing cerebrospinal fluid tau and *APOE* genotype between clusters we showed that it is plausible that clusters summarize genuine heterogeneity.

Some assumptions have been considered in the definition of Temporal Clustering, including symmetric and independent distribution of errors, as implied by the use of least squares estimation. Moreover, the bounded nature of MMSE, which takes a minimum of zero and a maximum of 30, means that the true distribution cannot be symmetric. In addition to this, Temporal Clustering assumes that data is missing completely at random, a stronger and less realistic assumption than the missing at random assumption of mixed models. However, even with these limitations it is encouraging that reasonable clustering accuracy was achieved in the simulation study after filtering for discrimination score, especially as we explicitly simulated missing data due to death.

While simulations show the effectiveness of Temporal Clustering at estimating cluster membership, they also show biased estimation of trajectory parameters, especially for clusters with a slow rate of decline (right panels

of Supplementary Figure S2.3). Reducing this bias could be useful in its own right, but may also improve the assignment of individuals to clusters. This bias can be due to overfitting of the individual offsets δ_i, which can be reduced in the future by penalizing unrealistic offsets. Alternatively it can be due to the mixture of between and within individual progression in the model, and the large extrapolation beyond the length of follow-up available (from ~ 2 years to 10 or even 50 years). Therefore, it is hard to know whether the ~ 65 year trajectory of the slowly declining cluster truly reflects within-individual change, or is an artefact of the model. This could be tested in a datasets with longer individual follow-up.

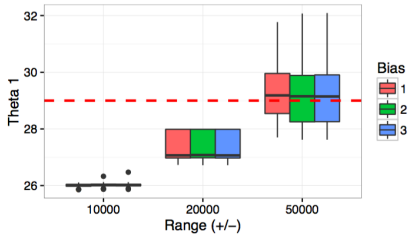
Despite the biased estimation of trajectory parameters, the more slowly declining cluster is still striking. From a clinical point of view this trajectory appears to decline too slowly to represent AD. Backing this up is the fact that it does include a higher proportion of individuals with Normal Condition or Mild Cognitive Impairment at last visit. However, the fact that around half of this cluster have a diagnosis of AD at last visit could suggest a problem with the model, perhaps motivating additional clusters. A less likely alternative hypothesis would be that individuals with a diagnosis of AD in the slowly declining cluster are misdiagnosed.

An advantage of mixed or random change point models over Temporal Clustering is that co-variates can be explicitly modeled, rather than considered post-hoc. Extending Temporal Clustering to consider co-variates could allow it to have more flexibility in the baseline of the model, this could get around the current crude assumption that the maximum MMSE in a lifetime is the same for all individuals.

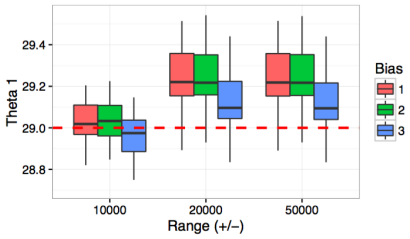
A limitation of this study is the use of MMSE to measure cognitive decline. MMSE is acknowledged to have ceiling and floor effects and to be relatively insensitive to cognitive change before Mild Cognitive Impairment of Proust-Lima et al. (2007). We concentrated on MMSE within this study as it is one of the most widely collected measures of cognitive ability in dementia. For example, longitudinal MMSE data is available for thousands of patients at the South London and Maudsley NHS Foundation Trust, where it has been extracted from Electronic Health Records from routine care by Perera et al. (2014). However, the method should be easily generalisable to other measures of cognitive ability.

In conclusion, we have demonstrated that it is possible to model cognitive decline using a combination of clustering and inference of individual offsets. This reduces, but does not eliminate, the effect of baseline MMSE on cluster assignment. Finally, we demonstrated a relationship between clusters and

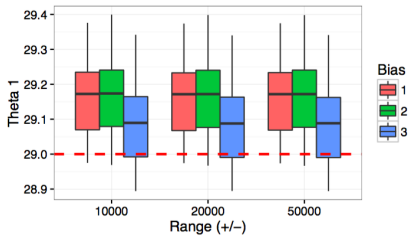
known AD risk factors. We believe that Temporal Clustering and future extensions will be useful for studying progression of dementia biomarkers.



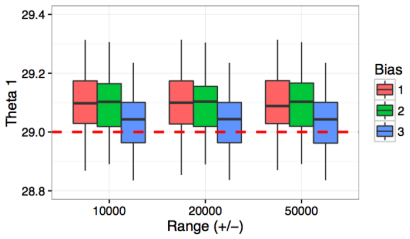
(A) Alignment simulation 1



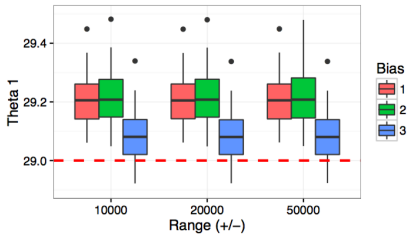
(B) Alignment simulation 2



(C) Alignment simulation 3

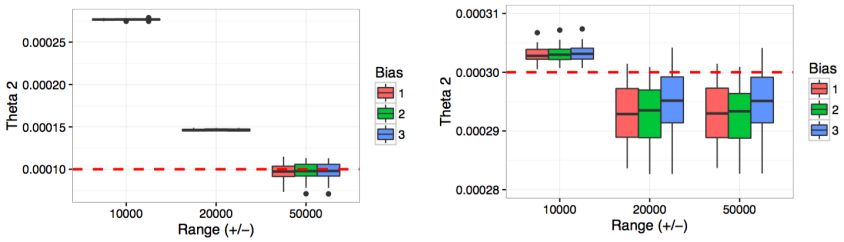


(D) Alignment simulation 4



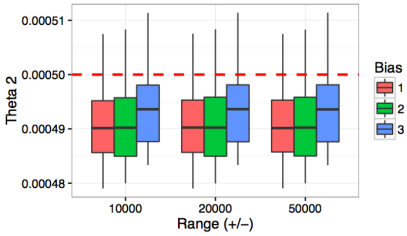
(E) Alignment simulation 5

SUPPLEMENTARY FIGURE S2.1: Grouped boxplots showing θ_1 estimates from the disease time estimation simulation. Five simulations each use a fixed true θ, with the true θ_1 indicated by dashed red lines. Each simulation differs only in the true value of θ_2: (A) 0.0001, (B) 0.0003, (C) 0.0005, (D) 0.0007 and (E) 0.0009. Simulations are based on the combined cohort, i.e. time points, number of samples and visual appearance. For alignment three different δ ranges were used: $\pm$ 10000, 20000 or 50000. Three versions of each dataset we generated, each with an increasing amount of bias and plausibility. Bias 1 = Real valued data, but with simulated death (i.e. MMSE $<$ 0 removed). Bias 2 = Integer rounded data with simulated death. Bias 3 = Integer rounded data with simulated death and a ceiling effect (MMSE $>$ 30 set to 30). Outliers excluded in plot for clarity: (B) 2 points $\sim$ 28, (D) 4 points $>$ 30.5 excluded and (E) 1 point $>$ 30.5 excluded.

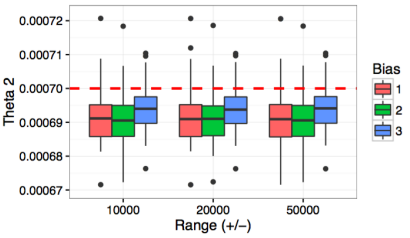


(A) Alignment simulation 1

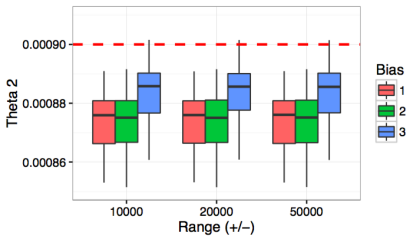
(B) Alignment simulation 2



(C) Alignment simulation 3

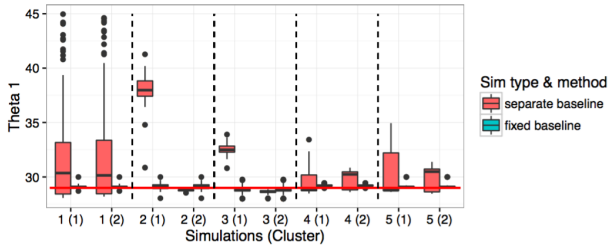


(D) Alignment simulation 4

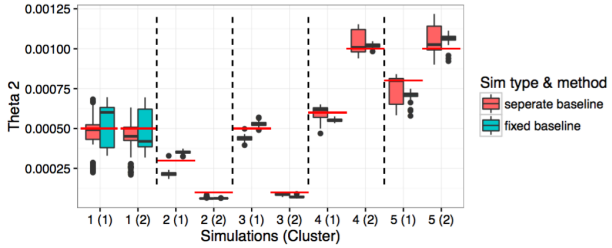


(E) Alignment simulation 5

SUPPLEMENTARY FIGURE S2.2: Grouped boxplots showing θ_2 estimates from the disease time estimation simulation. Five simulations each use a different fixed true θ. Each simulation differs only in the true value of θ_2, indicated by dashed red lines, as in Supplementary Figure S2.1. Simulations are based on the combined cohort, i.e. time points, number of samples and visual appearance. For alignment three different δ delta ranges were used: $\pm$ 10000, 20000 or 50000. Three versions of each dataset we generated, each with an increasing amount of bias and plausibility. Bias 1 = Real valued data, but with simulated death (i.e. MMSE $<$ 0 removed). Bias 2 = Integer rounded data with simulated death. Bias 3 = Integer rounded data with simulated death and a ceiling effect (MMSE $>$ 30 set to 30). Outliers excluded in plot for clarity: (B) 2 points $>$ 0.00032, (D) 4 points $\sim$ 0.0006 excluded and (E) 10 point $<$ 0.00085 excluded.



(A) Temporal clustering (K = 2) simulation - θ_1 estimates



(B) Temporal clustering (K = 2) simulation - θ_2 estimates

SUPPLEMENTARY FIGURE S2.3: Grouped boxplots showing the result of a simulation study of temporal clustering (K = 2). Each simulation differs only in the true value of θ_2^1 and θ_2^2. The results for each simulation are separated by dashed lines. Simulations are based on the combined cohort, i.e. time points, number of samples and visual appearance. True θ are indicated in red lines. Results are shown for both fixed and separate baseline versions of Temporal Clustering. As discussed in Section 2.4 the unfixed baseline lead to unrealistic estimates of the θ_1 parameter ($\theta_1 > 30$) and then this scenario is omitted in the following analyses.

Part II

FUNCTIONAL LINEAR MODELS

... we have been exploring the variability of a functional variable without asking how much of its variation is explainable by other variables. It is now time to consider the use of covariates... does the shape of the mean annual precipitation depend on which climate zone the station is? We need to see if the data reject the null hypothesis that there is no difference. And if this happens, then we want to characterize the differences in functional terms.

Ramsay and Silverman (2005)

This is how Ramsay and Silverman present the need for a model which takes into account the functional structure of outcomes with respect to categorical, or more broadly, scalar predictors. This is just one of the possible settings for functional linear models: responses, as well as covariates, can be either functions or scalar; but in this work we just focus on the Function-on-Scalar regression problem. Specifically, given a set of functions $y_1(t), \dots, y_N(t) \in \mathcal{H}$, where $\mathcal{H}$ a general Hilbert Space, and the correspondent set of predictors $\mathbf{x}_1, \dots \mathbf{x}_N \in \mathbb{R}^I$ we aim to estimate the functional influence $\beta_1(t), \dots, \beta(t)_I$ of the predictors on the functional outcomes, expressed as

$$y(t) = \sum_{i=1}^{I} \beta_i(t) x_i + \varepsilon(t),$$

where $\varepsilon(t)$ is a random noise. The key aspects of the analysis of functional linear models are the detection of the true set of relevant predictors among the ones introduced and the estimation of the contribution of these relevant predictors. Besides Ramsay and Silverman (2005), there are few recent works on the solution of Function-on-Scalar regression problems: Chen et al. (2016), for example, propose to combine functional least squares with a sparsity inducing penalty and to use a pre-whitening technique to exploit the within curve dependence. However, their method is computationally expensive and cannot be applied when the number of predictors, I, is greater than the sample size, N. Moreover, Barber et al. (2016) and Fan and Reimherr (2016) propose efficient methods to estimate the coefficients and induce sparsity in the model, with the introduction of a LASSO penalty; the second approach achieves also the functional oracle property, making the LASSO bias asymptotically negligible. In the model we propose here we firstly aim to identify the predictors, as the other methods do, but we then aim to control the smoothness level of the estimated coefficients. In the high dimensional setting ($I \gg N$) of genomic studies, in fact, the detection of the subset (possibly small) of the really effective predictors and the estimation of their influence

in such a way that the coefficients are smooth enough to be easily interpretable plays a key role.

In this Part we propose an innovative method to deal with variable selection and smoothing in high dimensional Function-on-Scalar regression with two applications to genomic data. This method is based on the solution of a LASSO penalty problem with the introduction of specific Hilbert spaces for the estimated parameters $\hat{\beta}(t)$. It is fully described in Chapter 3, where an application related to the influence of Single Nucleotide Polymorphysms on children affected by asthma is also presented. Moreover, a further real case study is introduced in Chapter 4. It is related to the identification of possible connections between the growth curves of children and the composition of their microbiomes.

Functional Linear Adaptive Mixed Estimation

In Chapter 3 we propose a method, called FLAME, to exploit simultaneously the smoothness of the estimation and the variable selection. None of the previously introduced Function-on-Scalar regression techniques fully exploits the smoothness of the underlying parameters. FLAME, instead achieves these two goals. Specifically, the coefficients are embedded in an Hilbert space, $\mathbb{K}$ which can be different from the space of the data. Here we choose a Reproducing Kernel Hilbert Space, so that the identification of a proper kernel allows us to tune the smoothness of the estimators or their particular structure, as periodicity. Then, the estimation is based on a Lasso penalization to guarantee variable selection; the algorithm is based on a coordinate descent method and it is efficiently coded to guarantee computational power. In this chapter we present a global overview of the method, its properties with asymptotic theory and simulations to highlight its effectiveness over existing methods. Finally, an application is illustrated: it focuses on the inspection of the influence of Single Nucleotide Polymorphysms (SNP) on the lung development of children affected by asthma. A SNP is a variation in a single nucleotide that occurs at a specific position in the genome. Here we isolate a set of 10.000 SNPs and we aim to detect which of them influence the lung capacity of children and how their influence is expressed during time. We isolate some SNP and in particular a polymorphysm on the MACROD2 gene which was not detected by other methods, but with a relevant biological interest, since it has already been connected with the asthma disease.

This Chapter is part of an already submitted work developed with Professor Matthew Reimherr from the Statistics Department of Penn State University and codes are available on the `flm` R package (that will be soon available on CRAN). The method, due to its sparsity inducing estimation, fits very well the genomic data requirements, but also finance or geosciences can provide interesting dataset to estimate the smooth impact of covariates on rough longitudinal measurements.

Microbiome and growth crurves

In Chapter 4 we apply the FLAME methodology previously introduced to connect the infant weight gain with the child and mother microbiome. Specifically, rapid infant weight gain has recently been associated with childhood obesity across the lifecourse (Monteiro and Victora (2005)). Moreover the microbiome is emerging as a causative environmental factor to the development of obesity (Hartstra et al. (2014)) and several studies have shown characteristic disturbances in obese adult and adolescent microbiomes when compared to normal weight peers. However, less is known about the establishment of the microbiome in early life and the effects of this early microbiome on weight. Therefore, we investigated the relationship between infant weight gain and gut and oral microbiome composition at age of 2 years. We collected data on the growth of children in their first years of age and three microbiome samples, one from the mother and two from children at their 2 years of age. With these data we aim to identify the impact of these microbiomes on the growth of children, computed as the ratio between weight and height. We identify the set of relevant bacteria in the three samples and analyze their impact on the growth curves of young children, isolating some bacteria of the Firmicutes and Bacteroidetes phyla causing an increment of the weight/height curve. This is a joint ongoing work with Professor Francesca Chiaromonte, Professor Matthew Reimherr from the Statistics Department of Penn State University, Professor Kateryna Makova, her Biology Lab at Penn State University and the Galaxy (Afgan et al. (2016)) team.

3

FLAME: SIMULTANEOUS VARIABLE SELECTION AND SMOOTHING FOR HIGH DIMENSIONAL FUNCTION-ON-SCALAR REGRESSION

3.1 INTRODUCTION

High-dimensional regression and functional data analysis are currently central research areas in statistics and machine learning. The rising interest in both areas reflects the difficult realities of *big data* that many scientists are now facing in their work. Increasingly complex studies and data gathering technologies require sophisticated methods which are at the same time mathematically sound, computationally efficient, and practically interpretable. This work concerns a new approach for function-on-scalar regression when the number of predictors is much larger than than number of statistical units. Such data is especially motivated by genetic studies where one encounters large numbers of scalar predictors. Such studies are also now increasingly likely to contain sophisticated phenotypic measurements that are suitable for functional data analysis. Our methodology simultaneously exploits the smoothness of the underlying data and functional parameters, as well as the sparsity of the genetic effects. For short, we call this framework FLAME, for Functional Linear Adaptive Mixed Estimation. The *mixed* here refers to the mixing of functional norms to simultaneously select significant predictors and smooth their corresponding effect on the functional outcome.

Currently, very little work has been done in this area, but there are several key recent papers which have made substantial in roads into this problem. For scalar-on-function regression, there are a few recent works Matsui and Konishi (2011), Lian (2013), Gertheiss et al. (2013), Fan et al. (2015), but this is the opposite of the problem we consider here. For funtion-on-scalar regression, Chen et al. (2016) propose to combine functional least squares with a sparsity inducing penalty. There they take the penalty to be the *group*

minimax concave penalty, MCP Zhang (2010). In addition, the authors use a pre-whitening technique to fully exploit the within curve dependence. Unfortunately, the method is computationally expensive and cannot be applied when the number of predictors, I, is greater than the sample size N, meaning that it cannot be applied to our intended high-dimensional applications. As we shall see in Section 3.3.2, the pre-whitening can also be counter productive when working with densely sampled functional data. Barber et al. (2016), instead, propose the function-on-scalar lasso, FSL, which also adds a penalty onto the functional least squares. In their approach they assume the data and parameters are from an arbitrary Hilbert space, but to induce sparsity the penalty is taken to be a type of induced ℓ_1 norm on the product space of Hilbert spaces where the parameters and data lie. Their approach is computationally efficient since it is a convex optimization problem, and achieves optimal rates of convergence for the parameter estimates even when the number of predictors, I, is much larger than the sample size N ($I \gg N$). However, the method, like traditional lasso, does not achieve the functional oracle property due to a non-negligible asymptotic bias. To that end, in a follow up paper Fan and Reimherr (2016) develop an adaptive version, AFSL, which achieves what we call here the *strong functional oracle property*, which we discuss in further detail in Section 3.2.4. Furthermore, this method can be implemented at nearly the same computational cost as FSL.

No previous methods specifically control the smoothing of the parameter estimates; they focus primarily on selecting the important predictors. To that end, our proposed work selects and simultaneously smooths the estimates. To accomplish this, we assume that while the data may live in an arbitrary Hilbert space, the parameters live in a smaller subspace which is a reproducing kernel Hilbert space, RKHS. By choosing different kernels for the space, one can exploit different assumptions about the parameters, especially smoothness and periodicity. We then translate these ideas into a penalized functional least squares problem. As we shall see in Section 3.3, this approach not only smoothes the parameter estimates, it uses the assumed underlying smoothness of the parameters to assist in the variable selection. Thus, not only our estimates are more likely to be interpretable, but they can also outperform previous methods in terms of variable selection when the parameters are sufficiently smooth or if they have some other structure than can be exploited such as periodicity (see Section 3.3.2).

In addition to introducing and outlining our proposed methodology, we present an asymptotic justification for our method including convergence rates and an oracle property. We also establish a very fast computational

framework for implementing the discussed methods, which is currently faster than both the FSL and AFSL implementations. This framework is part of an accompanying R package, `flm`, whose backend is written in `C++`, meaning that researchers may readily use these methods in their own work.

This chapter is organized as follows. In Section 3.2.1 we outline several important concepts from FDA and Reproducing Kernel Hilbert Spaces as well as the modeling assumptions on the data. In Section 3.2.3 we detail our approach presenting the coordinate descent algorithm which allows FLAME to be computed very efficiently. In Section 3.2.4 we present asymptotic theory, and in Section 3.3 we present numerical simulations to compare FLAME with previous methods and, then, in Section 3.4 we present an application to a longitudinal genetic association study.

3.2 METHODS

3.2.1 *Functional linear models and RKHS*

Our theory holds quite generally for data from an arbitrary real separable Hilbert space. In this way, our methodology is quite broad covering typical spaces such as $L^2[0,1]$, as well as product spaces, Sobolev spaces, etc. Then let $\mathbb{H}$ be a real separable Hilbert space, with norm $\|\cdot\|_{\mathbb{H}}$, while let K be a compact linear operator from $\mathbb{H} \to \mathbb{H}$. We assume that it is positive definite and self-adjoint:

$$\langle Kx, x\rangle \geqslant 0 \qquad \langle Kx, y\rangle = \langle x, Ky\rangle.$$

The spectral theorem Dunford and Schwartz (1963) implies that we can decompose K as

$$K = \sum_{i=1}^{\infty} \theta_i v_i \otimes v_i,$$

where $\theta_1 \geqslant \theta_2 \geqslant \cdots \geqslant 0$ are the ordered eigenvalues and $v_i \in \mathbb{H}$ are the corresponding eigenfunctions. The eigenfunctions $\{v_i\}$ form an orthonormal basis in $\mathbb{H}$. The tensor product $x \otimes y$ is used to denote the operator $(x \otimes y)(h) := \langle y, h\rangle x$. Then we define a subspace of $\mathbb{H}$, denoted $\mathbb{K}$, as follows:

$$\mathbb{K} := \left\{ h \in \mathbb{H} : \sum_{i=1}^{\infty} \frac{\langle h, v_i\rangle^2}{\theta_i} = \langle K^{-1}h, h\rangle < \infty \right\}.$$

If we equip $\mathbb{K}$ with the norm $\|h\|_{\mathbb{K}} = \|K^{-1/2}h\|_{\mathbb{H}}$ then this space is also an Hilbert space. Here it is understood that $0/0 = 0$. When $\mathbb{H}$ is $L^2[0,1]$ and the kernel of K is a bivariate function, i.e. $K(t,s)$, then $\mathbb{K}$ is also a reproducing kernel Hilbert space (Berlinet and Thomas-Agnan (2011)).

Focusing on the linear regression model, we now make the following modeling assumption about the response functions, $Y_n \in \mathbb{H}$, and the predictors $X_{n,i} \in \mathbb{R}$.

Assumption 1 *Let $Y_1, \ldots, Y_N$ be elements of $\mathbb{H}$, satisfying the functional linear model*

$$Y_n = \sum_{i=1}^{I} X_{n,i}\beta_i^\star + \varepsilon_n,$$

where $\mathbf{X} = \{X_{n,i}\} \in \mathbb{R}^{N\times I}$ is the deterministic design matrix with standardized columns, and ε_n are i.i.d. Gaussian random elements of $\mathbb{H}$ with mean function 0 and covariance operator C. We furthermore assume that there exists $0 \leqslant I_0 \leqslant I$ such that only $\beta_1^\star, \ldots, \beta_{I_0}^\star$ are nonzero. This means that, for notational simplicity, the first I_0 of the predictors are significant in the model. We will use the notation $\mathbf{X} = (\mathbf{X}_1\ \mathbf{X}_2)$ to partition the predictors into the significant predictors, $\mathbf{X}_1$, and the null predictors $\mathbf{X}_2$.

Note that any Gaussian process in $\mathbb{H}$ has necessarily a mean function in $\mathbb{H}$ and a covariance operator C which is compact, symmetric, and positive definite (Laha and Rohatgi (1979)). In our theory, the normality is only used to derive functional concentration inequalities. These inequalities determine the rate at which I can grow with N. When the errors are Gaussian, I can grow exponentially fast relative to N and the assumptions (as given in Assumption 2) are easier to be interpreted. Our arguments can be readily generalized to the non-normal case, but the rates will change and the assumptions will be more complicated, we thus do not pursue that direction presently.

3.2.2 *FLAME: the choice of the kernel*

The FLAME target function is given by

$$L(\beta) = \frac{1}{2N}\sum_{n=1}^{N}\|Y_n - X_n^\top\beta\|_{\mathbb{H}}^2 + \lambda\sum_{i=1}^{I}\tilde{\omega}_i\|\beta_i\|_{\mathbb{K}} = \frac{1}{2N}\|Y - \mathbf{X}\beta\|_{\mathbb{H}}^2 + \lambda\sum_{i=1}^{I}\tilde{\omega}_i\|\beta_i\|_{\mathbb{K}},$$

with $Y \in \mathbb{H}^N$, $\mathbf{X} \in \mathbb{R}^{N\times I}$ and $X_n = \mathbf{X}_{(n,\cdot)} \in \mathbb{R}^I$, $\beta \in \mathbb{K}^I$. Throughout, we use notation such as $\mathbb{H}^N$ to denote product spaces. For the sake of simplicity, we abuse notation a bit by letting $\|\cdot\|_{\mathbb{H}}$ also denote the induced Hilbert space norm on product spaces such as $\mathbb{H}^N$. There at least a few data driven ways one can choose the weights $\tilde{\omega}_i$. One way would be to use marginal regressions to get initial parameter estimates, then the weights would be one over the norms of those estimates (Huang et al. (2008)). Another option is to run FSL first and then use its corresponding estimates. This has the advantage of also dramatically reducing the dimension of the problem, and is the approach we take for developing our asymptotic theory in Section 3.2.4. Lastly, one could first run the nonadaptive version of FLAME (i.e. with $\tilde{\omega} \equiv 1$) to obtain preliminary estimates, $\tilde{\beta}_i$, and then compute the weights as $\tilde{\omega}_i = \|\hat{\beta}_{i,N}\|_{\mathbb{K}}^{-1}$. This is the approach we take for our empirical work in Section 3.3. Our reasoning is that we wanted a more pure comparison of the different methods to analyze their performances. Since all of the functional methods, except FSL, utilize a preliminary run to different degrees, opening the door to mixing and matching would create a huge number of potential options, which is beyond the scope of this work.

In our approach we use the norm $\|\cdot\|_{\mathbb{K}}$ to both induce sparsity and smooth the parameter estimates. Previous approaches have focused only on one or the other. Furthermore, by allowing for a general K, we provide a framework which is very flexible and can accommodate a variety of underlying assumptions about the parameters, such as periodicity and boundary conditions. The purpose of the data driven weights is to penalize "smaller" parameters more, and thus not shrink the larger ones as much. This allows the estimator to be asymptotically unbiased and achieve an oracle property. We now discuss several examples of popular kernels.

Example 1 (Sobelev Space) *Consider* $\mathbb{H} = L^2(\mathcal{D})$*, where* $\mathcal{D}$ *is a compact subset of* $\mathbb{R}^d$*. Define* $\mathbb{K}$ *to be the subset of functions in* $L^2(\mathcal{D})$ *that have up to and including* m^{th} *order derivatives that are also in* $L^2(\mathcal{D})$*. A family of norms can be defined on* $\mathbb{K}$ *as*

$$\|x\|_{\mathbb{K}}^2 = \sum_{|\alpha|\leqslant m} \frac{1}{\sigma_\alpha^2} \int_{\mathcal{D}} |x^{(\alpha)}(\mathbf{s})|^2 \, d\mathbf{s}.$$

Here α is a d-dimensional vector of integers whose sum is less than or equal to m, while the σ_α are nonzero weights. Equipped with this norm, $\mathbb{K}$ is an RKHS if and only if $m > d/2$. In the case where $\mathcal{D} = [0,1]$ and $m = 1$, we have that

$$K(t,s) = \begin{cases} \frac{\sigma}{\sinh(\sigma)} \cosh(\sigma(1-s)) \cosh(\sigma t) & t \leqslant s \\ \frac{\sigma}{\sinh(\sigma)} \cosh(\sigma(1-t)) \cosh(\sigma s) & t > s \end{cases}.$$

One can then numerically solve for the eigenfunctions and eigenvalues of K. These details can be found on Berlinet and Thomas-Agnan (2011).

Example 2 (Gaussian Kernel) *Let $\mathbb{H} = L^2(\mathcal{D})$, then the Gaussian kernel is given by*

$$K(\mathbf{s}, \mathbf{s}') = \exp\left\{-\sigma|\mathbf{s} - \mathbf{s}'|^2\right\}.$$

While the Sobelev spaces contain functions which are differentiable up to a given order, the space $\mathbb{K}$ here contains functions which are infinitely differentiable. When used in FLAME, such a kernel will produce very smooth estimates.

Example 3 (Exponential Kernel) *The exponential kernel is on the other end of the "smoothness" spectrum compared to the Gaussian kernel. In this case we have*

$$K(\mathbf{s}, \mathbf{s}') = \exp\left\{-\sigma|\mathbf{s} - \mathbf{s}'|\right\}.$$

This seemingly minor adjustment to the power in the exponent produces a space consisting of continuous functions which need not be differentiable. Using this kernel will produce substantially rougher FLAME estimates than the Gaussian kernel. In practice, they will also be a bit rougher than the Sobolev kernel as well.

Example 4 (Periodic Kernel) *A very useful feature of working with an RKHS is that one can incorporate structures such as periodicity and boundary conditions into the parameter estimates. This may be useful, for example, when the domain represents time over the course of a year. In that case, one might expect the parameters to be periodic. In this case one may use the periodic kernel with period $p = 1$ for yearly periodicity, $p = 1/2$ for semestral periodicity, or $p = 1/4$ for seasonal. The periodic kernel with period p is defined as*

$$K(\mathbf{s}, \mathbf{s}') = \sigma^2 \exp\left\{-2/\sigma \sin^2\left(\frac{\pi|\mathbf{s} - \mathbf{s}'|}{p}\right)\right\}.$$

More general boundary conditions can be worked into Sobelev spaces and norms, but we refrain from printing the details here, since we will not explore them in our

simulations. An interested reader is referred to, for example, Section 4 of Chapter 7 in Berlinet and Thomas-Agnan (2011) who list many examples of kernels that can work in different structures.

3.2.3 *FLAME: implementation and computational details*

In this section we develop a coordinate descent algorithm for quickly finding the FLAME estimator. These methods are implemented in an accompanying `R` package `flm`. The computationally intensive functions in this package are coded in `c++`, so the methodology can be computed very quickly even for very large datasets.

The algorithm is based on utilizing functional subgradients so that, at each step, individual parameter estimates can be updated very quickly in a nearly closed form. An interested reader is referred to Boyd and Vandenberghe (2004) Bauschke and Combettes (2011) Barbu and Precupanu (2012) Shor (2012) for more details on subgradients and subdifferentials. Subgradients generalize derivatives (in this case Fréchet derivatives) to convex functionals (mappings from $\mathbb{H}$ to $\mathbb{R}$) which are not necessarily differentiable. At any point where the functional is differentiable, the two notions coincide, but subgradients are well defined much more broadly to convex functionals that need not be differentiable. Let $f : \mathbb{H} \to \mathbb{R}$ be a convex functional. We say that $h \in \mathbb{H}$ is a subgradient of f at $x \in \mathbb{H}$ if for all $y \in \mathbb{H}$ we have

$$f(x+y) - f(x) \geqslant \langle h, y \rangle.$$

We denote by $\partial f(x)$ the collection of all subgradients of f at x, called the subdifferential. Trivially, x is a minimizer of f if and only if $0 \in \partial f(x)$. We show in the Appendix that the subgradient for FLAME is given by

$$\frac{\partial}{\partial \beta_i} L_\lambda(\beta) = -\frac{1}{N}\sum_{n=1}^{N} X_{n,i} K(Y_n - X_n^\top \beta) + \lambda \tilde{\omega}_i \begin{cases} \|\beta_i\|_{\mathbb{K}}^{-1}\beta_i, & \beta_i \neq 0 \\ \{h \in \mathbb{K} : \|h\|_{\mathbb{K}} \leqslant 1\}, & \beta_i = 0 \end{cases}. \tag{3.1}$$

At each step of the coordinate descent we can use (3.1) to update our estimates. In particular, suppose that $\hat{\beta}$ is our current estimate and we aim to update the i^{th} coordinate, $\hat{\beta}_i$. The least squares estimator would be

$$\check{\beta}_i = \frac{1}{N}\sum_{n=1}^{N} X_{n,i} E_n \qquad \text{where} \qquad E_n = Y_n - \sum_{j \neq i} X_{n,j}\hat{\beta}_j.$$

We can then express the subgradient as

$$\frac{\partial}{\partial \beta_i} L(\beta) = -K(\check{\beta}) + K(\beta_i) + \lambda \omega_i \begin{cases} \|\beta_i\|_{\mathbb{K}}^{-1} \beta_i, & \beta_i \neq 0 \\ \{h \in \mathbb{K} : \|h\|_{\mathbb{K}} \leqslant 1\}, & \beta_i = 0 \end{cases}.$$

We can immediately observe that

$$\|K(\check{\beta}_i)\|_{\mathbb{K}} \leqslant \lambda \omega_i \Rightarrow \hat{\beta}_i = 0. \tag{3.2}$$

Note this also indicates a useful starting point for the algorithm; if we take

$$\lambda = \max_{i=1,\dots,I} \{\omega_i^{-1} \|N^{-1} \sum X_{ni} K(Y_n)\|_{\mathbb{K}}\}, \tag{3.3}$$

then the solution will always be $\hat{\beta}_i = 0$. When $\hat{\beta}_i \neq 0$, we can solve for it in a nearly closed form. In particular, we have

$$-K(\check{\beta}_i) + K(\hat{\beta}_i) + \frac{\lambda \omega_i}{\|\hat{\beta}_i\|_{\mathbb{K}}} \hat{\beta}_i = 0 \quad \Rightarrow \quad \hat{\beta}_i = \left(K + \frac{\lambda \omega_i}{\|\hat{\beta}_i\|_{\mathbb{K}}} I \right)^{-1} K(\check{\beta}_i). \tag{3.4}$$

The only unknown quantity at this point is $\|\hat{\beta}_i\|_{\mathbb{K}}$. Unfortunately, its expression does not have a closed form solution (unlike FLS or AFSL). However, if we take the $\mathbb{K}$-norm of the expression in (3.4) we arrive at the following equation that can be solved numerically

$$1 = \sum_{j=1}^{\infty} \frac{\theta_j \langle \check{\beta}_i, v_j \rangle^2}{(\theta_j \|\hat{\beta}_i\|_{\mathbb{K}} + \lambda \omega_i)^2}.$$

Our coordinate descent algorithm therefore proceeds iteratively, defining a sequence of $\beta^{(t)}$ for $t = 1, \dots, T$ which converges to the desired approximation $\hat{\beta}$. We set the maximum number of iterations T and a stopping criteria based on the improvement in the estimation of the β coefficients (i.e. the $\mathbb{K}$-norm of the increment should be higher than a fixed tolerance).

Regarding the weights, $\tilde{\omega}_i$, we run the algorithm twice. The first one (*the non-adaptive step*) is run with weights set to 1, and the second time (*adaptive step*) we take $\tilde{\omega}_j = \|\hat{\beta}_{j,N}\|_{\mathbb{K}}^{-1}$ with $\|\hat{\beta}_{j,N}\|_{\mathbb{K}}$ the norm of the β estimated in the *non-adaptive step*. In particular, the *adaptive step* is run to improve the estimation of the meaningful predictors and then the algorithm is run only on the non-zero predictors isolated in the *non-adaptive step*. These steps must be

run for a sequence of λ and we have to identify a proper λ which maximizes some selection criterion; we choose λ to minimize the cross validation error, once we have isolated a training and a test set (randomly sampled as the 25% of the whole data set).

The complete schema of the algorithm is presented in Figure 3.1.

We mention two features we have built into the code which help increase its computational efficiency. The first is a *warm start* which means when moving to the next λ value, we use the previous $\hat{\beta}$ as the initial value for β. Since λ usually changes very little with each step, this means that the new $\hat{\beta}$ can be computed very quickly (usually with just a few iterations). In this way, one can obtain the solutions for an entire sequence of λ with only marginally more computation time than with a single λ. The second feature is what we call a *kill switch*. This allows the user to set the maximum size for the number of predictors selected by the model. When the algorithm moves past this threshold, the algorithm is stopped. In certain applications, one can make a good guess as to the maximum number of predictors that could conceivably be selected by the model. In these settings, this bound can be used for the *kill switch*. For example, in genetic studies, even with hundreds of thousands of predictors, it is usually safe to assume far fewer than say 100 SNPs, will actually be selected (usually the number is far less than 100). The algorithm slows down as more predictors enter the model, thus this has the potential to provide a substantial computational savings.

Lastly, all functional data methods of this type require some preprocessing of the raw data into functional units. This is now a fairly well developed step and a more detailed discussion can be found in Horváth and Kokoszka (2012). In short, we utilize a penalized cubic bsplines expansion, where the penalty is chosen by generalized cross validation. The number of bsplines in our simulations and application is taken to be 100 so that the smoothing is determined entirely by the penalty. In FSL and AFSL one would then commonly rotate to the FPCA basis so that less that 100 basis functions can be used, thus gaining computational efficiency. For FLAME, we instead use the eigenfunctions of the kernel K, which we compute numerically on a fine grid. This allows us to quickly compute both $\mathbb{H}$ norms and $\mathbb{K}$ norms. We choose the number of basis functions, J, so that

$$\sum_{j=1}^{J} \theta_j \geqslant 0.99 \sum_{j=1}^{\infty} \theta_j,$$

Algorithm 3: FLAME

Data: $\mathbf{X}, \mathcal{Y}, I_0, p_{cv}$, kernel (ν_j, θ_j), for $j = 1 : J$;
$I = \dim(\mathbf{X})[1]$ **begin** *Non-Adaptive step*
  $\omega_{NA} \leftarrow (1, \ldots, 1)$;
  $\Lambda_{NA} \leftarrow (\lambda_{NA}^{max}, \ldots, \lambda_{NA}^{min})$;
  for $\lambda \in \Lambda_{NA}$ **do**
    Update $\mathcal{B} \leftarrow$ Coordinate_descent$(\lambda, \mathbf{X}, \mathcal{Y}, \omega_{NA})$;
    if #*zeros*$(\mathcal{B}) > I_0$ **then**
      $\Lambda_{NA} \leftarrow (\lambda_{NA}^{max}, \ldots, \lambda)$;
      Break;
    end
  end
  $(\mathcal{B}_{NA}, \lambda_{NA}) \leftarrow$ Cross_validation$(\Lambda_{NA}, \mathbf{X}, \mathcal{Y}, p_{cv})$;
  $\mathcal{S}_0 \leftarrow$ indices of the non-zeros columns of $\mathcal{B}$;
end
begin *Adaptive step*
  $\omega_A \leftarrow 1/\|\mathcal{B}_{NA}[, \mathcal{S}_0]\|_{\mathbb{K}}$;
  $\Lambda_A \leftarrow (\lambda_A^{max}, \ldots, \lambda_A^{min})$;
  for $\lambda \in \Lambda_A$ **do**
    Update $\mathcal{B} \leftarrow$ Coordinate_descent$(\lambda, \mathbf{X}[, \mathcal{S}_0], \mathcal{Y}, \omega_A)$;
    if #*zeros*$(\mathcal{B}) > I_0$ **then**
      $\Lambda_A \leftarrow (\lambda_A^{max}, \ldots, \lambda)$;
      Break;
    end
  end
  $(\mathcal{B}, \lambda_A) \leftarrow$ Cross_validation$(\Lambda_A, \mathbf{X}[, \mathcal{S}_0], \mathcal{Y}, p_{cv})$;
end

FIGURE 3.1: Schema of the FLAME estimation: $\mathbf{X}$ is the design matrix, $\mathcal{Y}$ is the set of response functions $y(t)$ on the J eigenfunctions of the kernel, I_0 the *kill switch* parameter: the maximum number of non zero components to be identified, p_{cv} the proportion of the data to define the training-set and λ^{max} and λ^{min} defined as presented in Section 3.2.3.

where θ_j are the eigenvalues of the kernel K. This formulation is similar to explaining 99% of the variability in FPCA. We use such a high mark because dimension reduction is not our goal; we aim to approximate the data nearly exactly.

3.2.4 *FLAME: theoretical properties*

In this section we provide several theoretical properties of FLAME. While this theory provides a strong justification for using FLAME, there are still several interesting theoretical questions which remain open and will be discussed below. We begin by making the following assumption concerning the various terms in the model. Very similar assumptions can also be found in Fan and Reimherr (2016).

Assumption 2 *The regression problem satisfies the following.*

1. ***Minimum Signal:*** *Let* $b_N = \min_{i \in S} \|K(\beta_i^\star)\|_{\mathbb{K}}$*, then we assume the lower bounded*
$$b_N^2 \gg \frac{I_0^2 \log(I)}{N}.$$

2. ***Tuning Parameter:*** *The tuning parameter* λ *satisfies the following lower and upper bounds*
$$\frac{I_0^{1/2} \log(I)}{N} \ll \lambda \ll \frac{b_N}{\sqrt{I_0}\sqrt{N}}.$$

3. ***Design Matrix:*** *Let* $\hat{\Sigma}_{11} = N^{-1}\mathbf{X}_1^\top \mathbf{X}_1$*, be the design matrix for only the true predictors. We assume the minimum eigenvalue* $\sigma_{\min}(\hat{\Sigma}_{11})$ *and maximum eigenvalue* $\sigma_{\max}(\hat{\Sigma}_{11})$ *satisfy:*
$$\frac{1}{\nu_1} \leqslant \sigma_{\min}(\hat{\Sigma}_{11}) \leqslant \sigma_{\max}(\hat{\Sigma}_{11}) \leqslant \nu_1.$$

4. ***Irrepresentable Condition*** *Let* $\hat{\Sigma}_{21} = N^{-1}\mathbf{X}_2^\top \mathbf{X}_1$*, be the cross covariance between the null and true predictors. We assume that*
$$\|\hat{\Sigma}_{21}\hat{\Sigma}_{11}^{-1}\|_{op} \leqslant \phi < 1$$
with ϕ *a fixed scalar and* $\|\cdot\|_{op}$ *the operator norm.*

The first assumption is called minimum signal condition and indicates the minimum magnitude (of the signals) required for detecting the relevant pre-

dictors. Notice that this condition is placed on $\beta^\star$ relative to K, which means that if K wipes out a signal, the algorithm will not be able to detect it. The second condition concerns the rate for λ, and takes a fairly familiar form Barber et al. (2016); Fan and Reimherr (2016). Since our FLAME formulation normalizes the sum of squares by N, the λ needs to tend to zero. The lower bound indicates that it cannot go to zero too quickly, otherwise one cannot guarantee that all of the null predictors are dropped. Conversely, the upper bound actually indicates two things, first if λ goes to zero too slowly then some of the significant predictors may also be dropped. Second, the upper bound on λ also ensures the bias is asymptotically negligible for establishing an oracle property. The third condition on the design matrix simply says that the design matrix for the true predictors must be well behaved. This ensures that the oracle estimate as well as the FLAME estimate are well behaved when restricted to the set of true predictors. The last condition is interpreted as requiring that the true predictors and the null predictors are not too correlated. This condition is essentially necessary to obtain an oracle property, as in Zhao and Yu (2006).

Under these conditions, we can now state our primary theorem, which states that FLAME recovers the true support with probability tending to 1, and that its projections are asymptotically normal.

Theorem 1 *If the regression problem satisfies Assumptions 1 and 2, the solution of the FLAME problem, $\hat{\beta}$, asymptotically*

1. *has the same support of the true solution of the regression problem*

$$P(\hat{\beta} \overset{S}{=} \beta^\star) \to 1,$$

2. *and is equivalent to the Oracle estimator in the sense that, for any sequence $h_n = \{h_{i,n}\} \in \mathbb{K}^I$ that satisfies $\|h_n\|_{\mathbb{K}} \leqslant M_1$ and $\sum \|C^{1/2} h_{i,n}\|_{\mathbb{H}}^2 \geqslant M_2 > 0$ we have*

$$\frac{\sqrt{N}\langle \hat{\beta} - \beta^\star, h_n \rangle}{\sigma_n} \xrightarrow{\mathcal{D}} \mathcal{N}(0,1) \;\; \textit{where} \;\; \sigma_n^2 = \sum_{i \in \mathcal{S}} \sum_{j \in \mathcal{S}} \hat{\Sigma}^{-1}_{11;ij} \langle C^{1/2} h_i, C^{1/2} h_j \rangle.$$

The first part of the theorem is a fairly standard result; we are showing that our method is variable selection consistent. The second result shows that the estimators are consistent and asymptotically normal, but there is a serious caveats to this, namely the projections are normal only when projected onto

an element of $\mathbb{K}$, not $\mathbb{H}$. If the Y_n were finite dimensional, then the two would be equivalent, but not in the functional setting.

In the context of functional data, we call Theorem 1 a *week oracle property* because the normality occurs in the week topology (i.e. on projections). Our next result shows that one can actually obtain a stronger result, namely, that FLAME and oracle estimates are asymptotically equivalent in the *strong* topology. For this reason, we say that the following theorem is a *strong oracle property*. First let us define the oracle estimate, namely

$$\hat{\beta}_O = \{(\mathbf{X}_1^\top \mathbf{X}_1)^{-1}\mathbf{X}_1^\top Y, 0\},$$

where 0 a vector of zero functions of length $I - I_0$.

Theorem 2 *Suppose Assumptions 1 and 2 are satisfied, but that I_0 is fixed. Furthermore, assume there exists a $\delta > 0$ and a constant $0 < B < \infty$ such that for all $i \in S$*

$$\sum_{j=1}^{\infty} \frac{\langle \beta_i^\star, v_j \rangle^2}{\theta_j^{1+\delta}} \leqslant B < \infty.$$

If λ is such that

$$\lambda \ll \frac{b_N}{N^{1/2[1+1/(1+\delta)]}},$$

then one also has that

$$\sqrt{N}\|\hat{\beta} - \hat{\beta}_O\|_{\mathbb{H}} = o_P(1).$$

Notice that we have introduced slightly stronger assumptions to achieve a strong oracle property. In particular, we needed a more explicit assumption on the rate at which the coordinates of $\beta^\star$ decrease. If $\delta = 0$ this simply implies that $\beta^\star$ is in $\mathbb{K}$. Lastly, we require a tighter control of the λ which depends on how quickly the coordinates of $\beta^\star$ decrease. If the coordinates actually terminate (i.e. are zero) at a certain point or if they decrease exponentially fast, then our assumption on λ is the same as before. The assumption that I_0 is fixed allows us to simplify the results. Using our techniques it is possible to allow I_0 to grow, but we would need additional assumptions on the behavior of the trace of the errors with respect to the $\{v_i\}$ basis, and so do not pursue it here.

We believe that our results can be tightened, especially the additional assumptions needed to achieve Theorem 2. Maybe the major obstacle is obtaining a good control of $\|\hat{\beta}\|_{\mathbb{K}}$. This quantity shows up when updating via coordinate descent and when trying to control the bias of the FLAME estimate. However, unlike FSL, we do not have an explicit expression for this

quantity in terms of the least squares estimator. If one can obtain a tighter control of this quantity, it should be easier to relax the assumptions of Theorem 2. Lastly, it might be interesting to study the asymptotic properties of $\hat{\beta}$ under the $\mathbb{K}$ norm, instead of the $\mathbb{H}$ norm. For example, it might be of interest to study the estimated derivatives of the parameters. However, since this is a much stronger norm, clearly additional assumptions will be needed. Furthermore, the oracle estimate would not be the least squares estimator as this need not even live in the space $\mathbb{K}$. We thus believe there are many open and exciting questions concerning the behaviors of such functional estimators and their necessary assumptions.

3.3 SIMULATION STUDIES

In this section we introduce several simulation schemes to analyze the performances of FLAME with different RKHS (Section 3.3.1) and to compare this method with AFSL and MCP (Section 3.3.2). For all simulations we assume data in $L^2[0,1]$. The kernels we consider are three popular kernels, the Exponential, the Sobolev, and the Gaussian. Moreover, for the specific case of Section 3.3.2 we introduce the periodic kernel. In Figure 3.2, the first four eigenfunctions associated to the Exponential, the Sobolev, and the Gaussian kernel are plotted and the explained variance is shown. These three kernels show different structure and complexity; in Section 3.3.1 the consequences of the different smoothness levels required to functions embedded in these kernels are presented.

All simulations used 100 runs on a Intel quad-core i7 desktop with 8GB of ram with the vecLib linear algebra library of R and measured in terms of:

- *computational time*: median of the computational time (sec.) of the runs.
- *number of true positive predictors*: average number of correctly non-zero predictors identified (i.e. $\#\{i : \beta_i^\star \neq 0 \wedge \hat{\beta}_i \neq 0\}$).
- *number of false positive predictors*: average number of wrongly identified non-zero predictors (i.e. $\#\{i : \beta_i^\star = 0 \wedge \hat{\beta}_i \neq 0\}$).
- *prediction error*: average of the prediction error, both for data and derivatives,
  $\sum_{n=1}^{N} \| \mathbf{X}_n \beta^\star - \mathbf{X}_n \hat{\beta} \|_{L^2}$ and $\sum_{n=1}^{N} \| \mathbf{X}_n \beta^{\star\prime} - \mathbf{X}_n \hat{\beta}' \|_{L^2}$

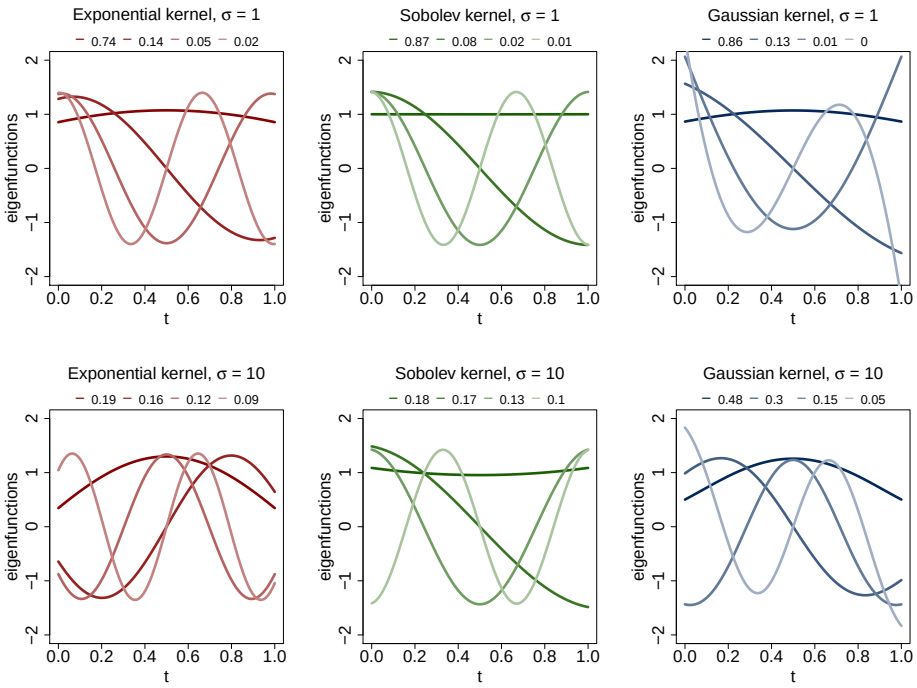


FIGURE 3.2: Representation of the first four eigenfunctions for each kernel with different σ. From the left: the Exponential, the Sobolev and the Gaussian kernel. The legend at the top of each panel denotes proportion of the explained variability for each eigenfunction.

3.3.1 *Comparison between different kernels*

In this section we compare the performance of FLAME using different kernels. We show how the variation of the kernel can influence the identification of the number of correctly identified predictors and the prediction error. Two high-dimensional simulation settings are introduced: with rough and smooth $\beta^\star$ coefficients.

The simulations consist of the random generation of a sample of size $N = 500$ and $I = 1000$ predictors, with $I_0 = 10$ significant ones. The predictor matrix $\mathbf{X}$ is the standardized version of a matrix randomly sampled from a N dimension Gaussian distribution with 0 average and covariance $\Sigma_X = \mathbb{1}$. For the rough case, the true coefficients $\beta^\star(t)$ are sampled from a Matérn process with 0 average and parameters $(\nu = 2.5, \text{range} = 1/4, \sigma^2 = 1)$, while for the smooth setting the range parameter of the Matérn process is set to 1 and ν is set to 3.5. In Figure 3.3 an example of the true coefficients in the two settings

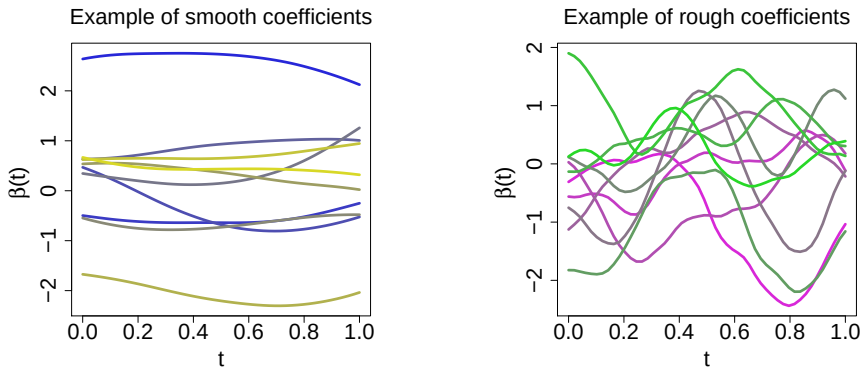


FIGURE 3.3: Example of 10 $\beta^\star$ coefficients for the smooth (left panel) and rough (right panel) simulation setting.

is shown. The outcomes, $Y_n(t)$, are obtained as the sum of the contribution of all the predictors and a random noise, a 0-mean Matérn process with parameters $(\nu = 1.5, \text{range} = 1/4, \sigma^2 = 1)$. Functions are sampled on an evenly spaced grid between 0 and 1 with $m = 50$ points.

For these simulations the *kill switch* parameter is set to $2I_0 = 20$ and λ spans a logarithmic equispaced 100-point gird from λ_{max} of Equation (3.3) to $r_\lambda \lambda_{max}$ with $r_\lambda = 0.01$ for the rough case and $r_\lambda = 0.001$ for the smooth setting. A summary of the result is shown in Figure 3.4 for the rough case and in Figure 3.5 for the smooth case.

Focusing on the rough setting we notice that the Gaussian kernel always performs worse than other kernels in terms of prediction error both for data and derivatives: it imposes on the functions a structure (infinitely differentiable) they don't possess. Moreover, increasing the σ parameter of the kernels, which results in a rougher estimates, reduces the prediction error and more true non zeros predictors are identified. In fact, with a too strong smoothness level, imposed by the Gaussian kernel or by a small value for the σ parameter, some true predictors are forced to be zero throughout the domain, reducing the number of true positives and increasing the prediction error. The rough structure of the parameters allows to all the methods presented to avoid the identification of non significant predictors and to keep the number of False Positive at zero.

A slightly different behavior can be observed in the smooth case. The performance of the Gaussian kernel, while still worse, is now much closer in performance to the other two kernels. The strange behavior of the prediction error of derivatives for the gaussian and the exponential kernel is due to

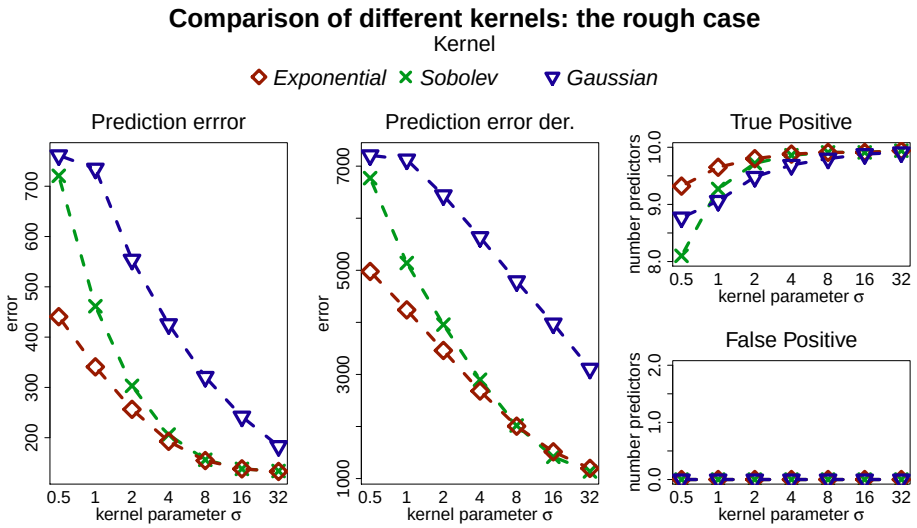


FIGURE 3.4: Summary of the simulations varying kernel for the rough case. From the left, the prediction error, the prediction error on derivatives, and the number of true and false positive predictors. We can notice that in all the simulations the number of False Positive estimated predictors is 0. No extra parameters are estimated with FLAME, while the number of True Positive predictors increases with the roughness level of the kernel.

an instability in the estimation of the derivatives of the eigenfunctions of these kernels at the boundaries of the time domain (not shown here). The number of False Positive predictors in this setting is different from zero (but it remains on average smaller than one per simulation).

A final remark regarding the high dimensional setting is the computational cost of the estimation and variable selection procedure. As presented in Table 3.1, the computational time is almost invariant with respect to the kernel and parameter, while increasing the smoothness level of the predictors increases the computational time. In the next section we present how FLAME is competitive compared to previous methods.

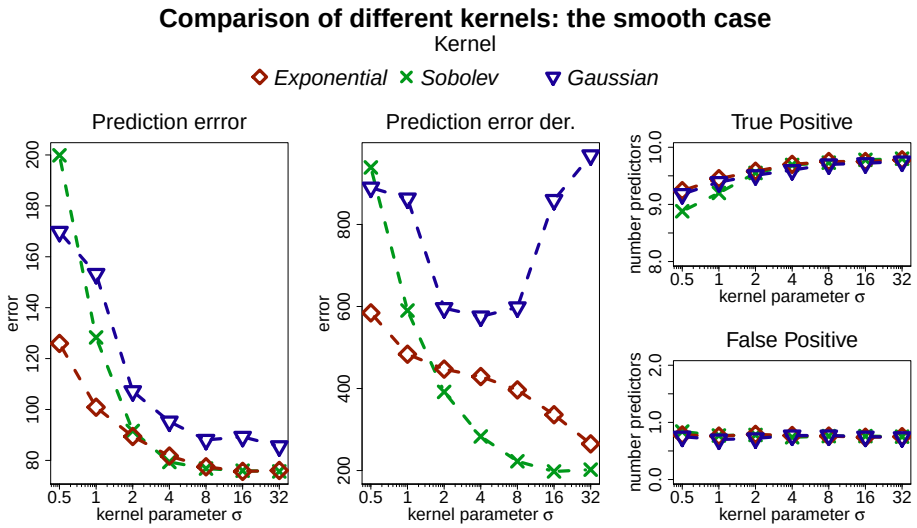


FIGURE 3.5: Summary of the simulations varying kernel for the smooth case. From the left, the prediction error, the prediction error on derivatives, and the number of true and false positive predictors.

3.3.2 *Comparison with previous methods*

The high dimensional setting

In this section we apply AFLS to the simulation setting we've introduced in Section 3.3.1 and in Table 3.2 we present the results of AFSL estimation in terms of prediction error, computation time and number of predictors identified (True Positive and False Positive).

A great advantage of FLAME is the reduction of the computation time: FLAME takes much less than AFSL to run and it also achieves better statistical performance. Mainly in the rough case, the Exponential and the Sobolev kernel (with $\sigma > 1$) perform better in terms of prediction error on data, derivatives and in the number of true positive and false positive predictors.

The small dimensional setting

In this section we reduce the simulation size to make the application of MCP possible; this method is suitable just for $N > I$ schemes. We present the results of FLAME, MCP, and AFSL with the same rough and smooth

FLAME			
	Kernel		
σ	Gaus.	Sob.	Exp.
0.5	29.30	30.75	41.14
1	21.64	36.62	48.17
2	28.87	43.92	58.67
4	32.34	39.14	61.48
8	32.61	42.99	47.29
16	33.67	42.59	39.95
32	35.47	33.47	40.83

FLAME			
	Kernel		
σ	Gaus.	Sob.	Exp.
0.5	80.38	85.81	95.00
1	77.67	81.33	94.66
2	72.23	87.59	97.95
4	66.69	76.18	91.18
8	58.46	79.12	99.08
16	61.14	80.36	92.98
32	63.23	70.22	69.97

TABLE 3.1: Median time for the simulations varying kernel for the rough (left panel) and smooth case (right panel).

	prediction error	prediction error der.	True Positive	False Positive	Time (sec.)
rough setting	352.51	4664.2	9.92	0.08	1031.01
smooth setting	95.43	382.17	9.64	0.41	812.24

TABLE 3.2: AFSL results for the rough and smooth high-dimensional simulation setting. Prediction error, computation time and number of correctly and wrongly identified predictors are presented. This results have to be compared with Figure 3.4 and 3.5 for the estimation and with Table 3.1 for the computational efficiency.

settings introduced in Section 3.3.1, but with $N = 50$, $I = 20$ and $I_0 = 5$. Moreover, we focus on the number of points per curve m to detect whether these three methods are affected by m. For FLAME we focus on the Sobolev kernel with $\sigma = 8$, since, from Section 3.3.1, it is shown to be a suitable kernel for both these two settings.

In Figure 3.6 and 3.7 the results for the three methods varying m are shown. We notice that both FLAME and AFSL estimations are almost invariant with respect to m, while MCP is strongly affected by variations of m, becoming very unreliable when the number of points per curve is large. However, if the number of points is small, MCP performs better than FLAME and AFSL in terms of prediction error and selecting true predictors, mainly in the smooth setting, but still has often trouble in terms of false positives. Focusing on the computational efficiency presented in Table 3.3, we notice that FALME and AFSL are comparable, with the well known higher efficiency of FLAME in the rough case with respect to the smooth, and they both are almost invariant with the change of m. They globally perform significantly better than MCP, which in addition becomes slower and slower with the increase of m. The difference in the efficiency of FLAME and AFSL is due to the method used to solve the problem: the coordinate descent method of FLAME is faster than ADMM of AFSL in the high dimensional

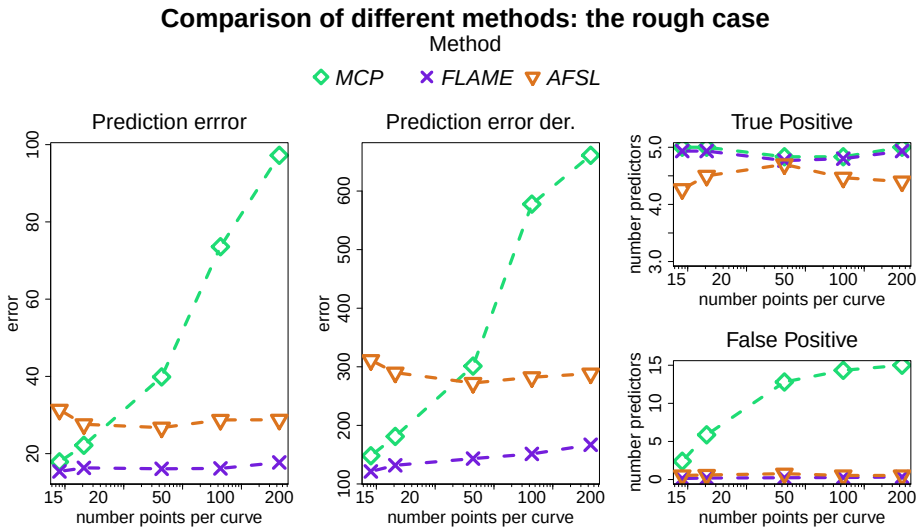


FIGURE 3.6: Summary of the simulations varying the method for the rough case. From the left, the prediction error, the prediction error on derivatives, and the number of true and false positive predictors.

setting since it is not based on matrix algebra operations, while in the small setting both coordinate descent and ADMM are efficient.

The periodic setting

In this section we focus on a distinctive feature of FLAME: the possibility of adapting the choice of the kernel to the prior knowledge on the data. For example in Figure 3.8 we plot several periodic coefficients $\beta^\star$. When using FLAME with a periodic kernel, the resulting estimates will also be periodic. In Figure 3.9, for example, the eigenfunctions of the periodic kernel

m	MCP	FLAME	AFSL
15	36.00	12.90	7.34
20	32.20	12.56	7.20
50	92.35	13.00	7.28
100	126.58	12.08	7.15
200	377.36	13.95	6.54

m	MCP	FLAME	AFSL
15	12.84	76.85	7.75
20	13.89	60.39	6.92
50	66.30	45.106	8.11
100	139.86	92.57	7.00
200	221.36	85.45	6.14

TABLE 3.3: Median time (sec.) for the simulations varying method for the rough (left panel) and smooth case (right panel) in the small dimensional setting.

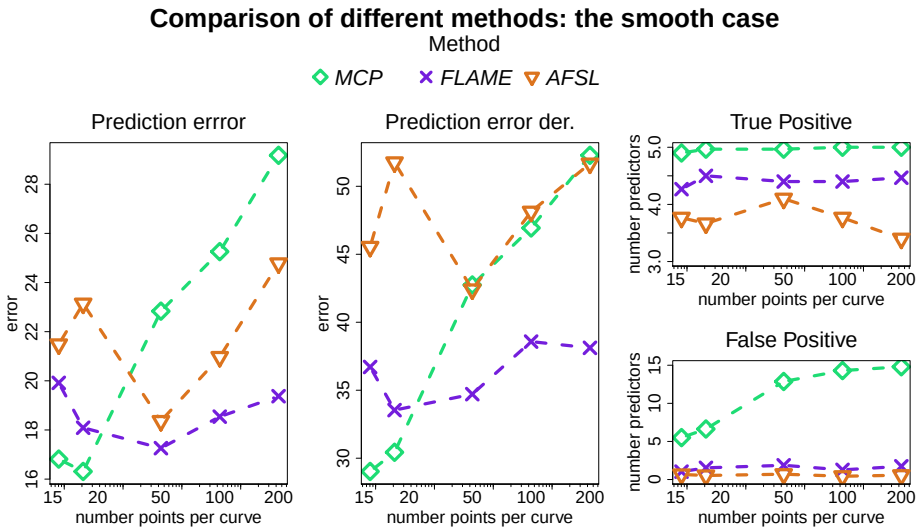


FIGURE 3.7: Summary of the simulations varying the method for the smooth case. From the left, the prediction error, the prediction error on derivatives, and the number of true and false positive predictors.

with period $1/2$ are shown. This kernel is general enough to be used for the estimations in a simulation setting where $\beta^\star$ functions are sampled as periodic functions with period varying in $\{1/2, 1/4, 1/8\}$. AFSL and MCP, on the contrary, don't allow this characterizations of the coefficients.

The design matrix $\mathbf{X}$ is the standardized realization of a multivariate normal distribution with 0 average and identity covariance structure and the errors are sampled from a Matérn process with parameter $(\nu = 1.5, \text{range} = 1/4, \sigma^2 = 1)$. The aim is to compare the results of FLAME, MCP, and AFSL. In this particular case, a kernel with period $\{1/2\}$ allows FLAME to estimate all the predictors identifying also their periodicity. MCP and AFSL, in contrast, are estimated in the general L^2 space, without any further specifications. In Table 3.4 we present a summary of the average results across 100 replications for the three methods where we see a fairly dramatic increase in statistical performance for FLAME. An example of the estimates produced by the different methods, based on $\beta^\star$ from Figure 3.8, is given in Figure 3.10, where we see a again a fairly dramatic advantage when using FLAME.

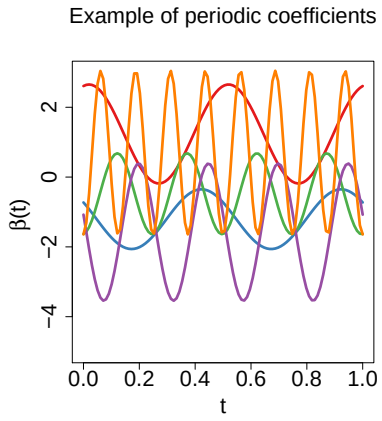


FIGURE 3.8: Example of 5 $\beta^\star$ periodic coefficients, two have period 0.5, two 0.25 and one 0.125.

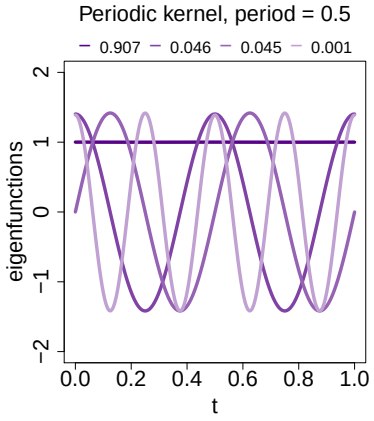


FIGURE 3.9: First four eigenfunctions of the periodic kernel with period 0.5. Correspondent explained variability is shown in the top legend

	prediction error	prediction error der.	True Positive	False Positive	Time
FLAME	24.99	666.15	4.93	0.03	25.99
MCP	162.24	4055.37	5	5	924.98
AFSL	54.54	2081.90	4.87	0.53	8.04

TABLE 3.4: Comparison of the results of the three methods on simulations in the periodic setting. Average prediction error on data, derivatives, average number of true positive, false positive and the median computational time are shown.

3.4 A REAL CASE STUDY: CAMP

In this section we present the application of FLAME to a large genetic dataset collected from The Childhood Asthma Management Program Research Group (1999). The Childhood Asthma Management Project, CAMP, is a longitudinal trial to analyze the longterm impacts of several daily treatments on children with asthma. It includes 439 Caucasian children, ages 5-12, affected by asthma and monitored for 4 years. These data are freely available from the dbGaP, Study Accession phs000166.v2.p1 (dbGaP (2009)).

Genotypic informations consists of approximately $670,000$ SNPs with minor allele frequency larger than 5%. We first apply a screening tool from Chu et al. (2016) to isolate a subset of $I = 10,000$ SNPs, on which we apply FLAME. The focus of our analysis is, then, the detection of the significant SNPs among these $10,000$.

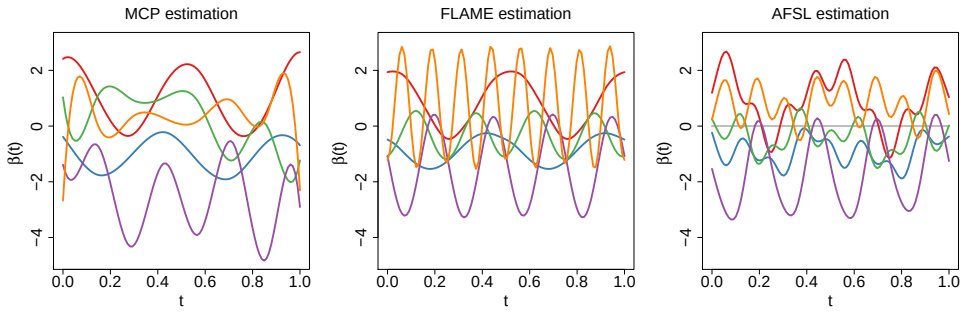


FIGURE 3.10: Example of the estimation of the functions of Figure 3.8 with, from the left, MCP, FLAME and AFSL.

Each child is given one of three treatments: Budesonide, Nedocromil, or a placebo. We account for age at the beginning of the study and gender. To quantify the lung strength of children we consider 16 longitudinal measurements of the Forced Expiratory Volume in one second (FEV1), a common proxy for lung strength. The lung capacity is the response function of our linear model and we convert it into a functional data object with a cubic Bspline basis projection with penalty on the second derivative and smoothing parameter chosen via generalized cross-validation.

As a first preprocessing step we remove the influence of gender, age, and treatment from FEV1 and then we apply FLAME to evaluate the impact of the SNPs to the residual functions shown in Figure 3.11. In Figure 3.12 the FLAME estimation is presented; for this analysis we use the Sobolev kernel with $\sigma = 8$, a 200 points grid for λ with the ratio $r_\lambda = 0.01$. We identify the presence of 12 significant SNPs, 9 with a positive effect in the lung development and 3 (rs2206980, rs2041420 and rs953044) with a negative contribution. In Table 3.5 the list of the identified SNPs with the comparison with the ones identified by AFSL: we notice that FLAME identifies two more SNPs, one with positive effect (rs722490) and one with negative effect (rs2041420).

To introduce a further comparison with AFSL we identify a test (made up by 80% of data) and a training set to compute the prediction error of data as $\sum_{n=1}^{N} \| Y_n - \mathbf{X}_n \hat{\beta} \|_{L^2}$. We replicate this analysis 10 times to present a robust conclusion. The average prediction error for FLAME is 0.200, while for AFSL is 0.205. Moreover, measuring the computational time we have for FLAME a median of 172.01 sec. and for AFSL 365.07 sec. showing the great advantage of FLAME in terms of computational time, with also a little improvement in term of prediction error.

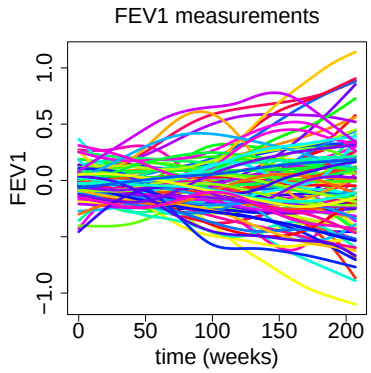


FIGURE 3.11: FEV1 curves of 100 randomly selected children measured on 4 years of follow up. The contribution of age, gender and treatment have already been removed.

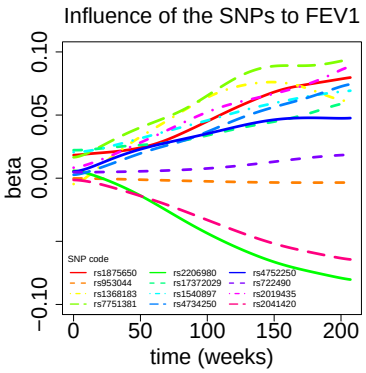


FIGURE 3.12: Coefficients of the influent SNPs detected and estimated by FLAME.

SNP		AFSL	FLAME
chr	name		
1	rs1875650	+	+
2	rs953044	-	-
5	rs1368183	+	+
6	rs7751381	+	+
6	rs2206980	-	-
7	rs17372029	+	+
8	rs1540897	+	+
8	rs4734250	+	+
10	rs4752250	+	+
11	rs722490		+
15	rs2019435	+	+
20	rs2041420		-

TABLE 3.5: List of the identified SNPs with AFSL and FLAME. + identifies the SNPs with positive effect and - the SNPs with negative effect, empty cells identify non detected SNPs. Informations on the chromosome location of SNPs and further details can be found in the ALFRED database (Rajeevan et al. (1999)).

As a last point, the SNP selected by FLAME but not by AFSL, rs2041420, is located on the gene MACROD2. This gene has been associated with a number of negative health outcomes including Autism, Celiac disease, Crohn's disease, and Parkinson's disease (`http://www.gwascentral.org`). It has also been linked to FEV1 and lung development Repapi et al. (2010); Strachan et al. (2007). However, neither of these previous studies were based on CAMP, and therefore helps validate this finding.

3.5.1 *The Subgradient Equation for FLAME*

Before deriving (3.1) we state the following Lemma which can found in any of the discussed references on convex analysis.

Lemma 1 *Let $f_1 : \mathbb{H} \to \mathbb{R}$ be $f_2 : \mathbb{H} \to \mathbb{R}$ be two convex functionals over a real separable Hilbert space $\mathbb{H}$. Then we have the following.*

1. *If the Fréchet derivative of f_1 exists at a point $x \in \mathbb{H}$, then the subdifferential of f_1 at x consists of single point which is the derivative of f_1 at x.*

2. *The subdifferential of $f_1 + f_2$ is the sum of their respective subdifferentials: $\partial(f_1 + f_2) = \partial f_1 + \partial f_2$. Where the sum is understood as Minkowski sum between two sets.*

We now state two lemmas from Fan and Reimherr (2016)

Lemma 2 1. *Consider the functional $f(x) = \|x\|_{\mathbb{H}}^2$. Then f is convex and everywhere differentiable with*

$$\partial f(x) = 2x.$$

2. *Consider the functional $f(x) = \|x\|_{\mathbb{H}}$. Then f is convex and differentiable when $x \neq 0$ in which case*

$$\partial f(x) = \|x\|_{\mathbb{H}}^{-1} x \qquad x \neq 0.$$

When $x = 0$ we have

$$\partial f(0) = \{x \in \mathbb{H} : \|x\| \leqslant 1\}.$$

We now derive the FLAME subgradient equations. First, we rewrite them using a common norm:

$$L_\lambda(\beta) = \frac{1}{2N}\|K^{1/2}(Y - \mathbf{X}\beta)\|_{\mathbb{K}}^2 + \lambda \sum_{i=1}^{I} \tilde{\omega}_i \|\beta_i\|_{\mathbb{K}}.$$

So L_λ is a convex function from $\mathbb{K}^I \to \mathbb{R}$. Here it is also understood that $K^{1/2}(Y)$ is applied coordinate wise to each function. Since $\mathbb{K}$ is a real separable Hilbert space we have by Lemma 2.1 and the chain rule that

$$\frac{\partial}{\partial \beta_i} \frac{1}{2N} \|K^{1/2}(Y - \mathbf{X}\beta)\|_{\mathbb{K}}^2 = \frac{1}{N} \sum_{n=1}^{N} X_{n,i}(K^{1/2}(Y - \mathbf{X}\beta)).$$

By Lemma 2.2 we have that

$$\frac{\partial}{\partial \beta_i} \lambda \sum_{j=1} \tilde{\omega}_j \|\beta_j\|_{\mathbb{K}} = \lambda \tilde{\omega}_j \begin{cases} \|\beta_j\|_{\mathbb{K}}^{-1} \beta_j & \beta_j \neq 0 \\ \{h \in \mathbb{H} : \|h\|_{\mathbb{K}} \leqslant 1\} & \beta_j = 0 \end{cases}.$$

Applying Lemma 1 we can combine the two subdifferentials to obtain (3.1).

3.5.2 *The week oracle property: Theorem 1.1*

The following two lemmas follow from Barber et al. (2016).

Lemma 3 *If Assumption 2 holds, the FSL estimate $\tilde{\beta}$, computed with all the weights set to 1, satisfies*

$$sup_{i \in \mathcal{S}} \|\beta_i^\star - \tilde{\beta}_i\|_{\mathbb{H}} = O_P(r_N^{1/2}) \qquad where \qquad r_N = \frac{\log(I) I_0}{N}.$$

Lemma 4 *Let X be an $\mathbb{H}$ valued Gaussian process with mean zero and covariance operator C. Then we have the bound*

$$P\left\{ \|X\|_{\mathbb{H}}^2 \geqslant \|C\|_1 + 2\|C\|_2 \sqrt{t} + 2\|C\|_\infty t \right\} \leqslant \exp(-t)$$

where $\|C\|_1$ the sum of the eigenvalues of C, $\|C\|_2^2$ the sum of the squared eigenvalues and $\|C\|_\infty$ the largest one.

Corollary 1 *Given the Gaussian process X, with zero mean and covariance operator C, and given the kernel operator K (represented by the eigenvalues θ_j: $\theta_1 \geqslant \theta_2 \geqslant \ldots \geqslant 0$, and the eigenvectors v_j which define an orthogonal basis for $\mathbb{H}$ and $\mathbb{K}$), we can prove that*

$$P\left\{ \|K(X)\|_{\mathbb{K}}^2 \geqslant \theta_1 (\|C\|_1 + 2\|C\|_2 \sqrt{t} + 2\|C\|_\infty t) \right\} \leqslant \exp(-t)$$

PROOF From the definition of the $\mathbb{K}$ and $\mathbb{H}$ norm we obtain that

$$\|K(X)\|_{\mathbb{K}}^2 = \sum_{j=1}^{\infty} \frac{\langle \theta_j X, v_j \rangle^2}{\theta_j} = \sum_{j=1}^{\infty} \theta_j \langle X, v_j \rangle^2 \leqslant \theta_1 \sum_{j=1}^{\infty} \langle X, v_j \rangle^2 = \theta_1 \|X\|_{\mathbb{H}}^2$$

Then, since from Lemma 4 we have that

$$P(\|X\|_{\mathbb{H}} < f(C,t)) \geqslant 1 - \exp(-t)$$

with $f(C,t) = \|C\|_1 + 2\|C\|_2\sqrt{t} + 2\|C\|_\infty t$.
we prove the statement

$$P(\|K(X)\|_{\mathbb{K}} < \theta_1 f(C,t)) \geqslant 1 - \exp(-t) \quad \Rightarrow \quad P\left(\|K(X)\|_{\mathbb{K}} \geqslant \theta_1 f(C,t)\right) \leqslant \exp(-t)$$

■

We begin by partitioning the set of the estimated parameters into $\hat{S}$ and $\hat{S}^C$ where

$$\hat{S} = \left\{ i \in \{1, \ldots, I\} : \hat{\beta}_i \neq 0 \right\}.$$

Our aim for this section is then to prove that, with high probability, $S = \hat{S}$, that is $\hat{\beta}$ has S as support.

Suppose, for the moment, that $\hat{S} = S$, then from the subgradient equation (3.1) we have that

$$\mathbf{X}_1^\top K\left(Y - \mathbf{X}_1 \hat{\beta}_1\right) = \lambda \tilde{s}_1 \qquad \text{where} \qquad \tilde{s}_1 = \left\{ N\tilde{\omega}_i \hat{\beta}_i \|\hat{\beta}_i\|_{\mathbb{K}}^{-1} : \ i \in S \right\}, \quad (3.5)$$

and $\hat{\beta}_1 = \{\hat{\beta}_i : \ i \in S\}$ is the estimate of the non-zero predictors. This then implies that

$$K(\hat{\beta}_1) = \left(\mathbf{X}_1^\top \mathbf{X}_1\right)^{-1} \left(\mathbf{X}_1^\top K(Y) - \lambda \tilde{s}_1\right) = K(\beta_1^\star) + \left(\mathbf{X}_1^\top \mathbf{X}_1\right)^{-1} \left(\mathbf{X}_1^\top K(\varepsilon) - \lambda \tilde{s}_1\right).$$

To prove that $\beta^\star$ and $\hat{\beta}$ have the same support ($S = \hat{S}$) we have to verify the following.

- If $i \in S$, $\hat{\beta}_1 \overset{S}{=} \beta_1^\star$, i.e. the true non-zero predictors are correctly identified. This condition can be also written as

$$\|K(\beta_i^\star) - K(\hat{\beta}_i)\|_{\mathbb{K}} < \|K(\beta_i^\star)\|_{\mathbb{K}}. \quad (3.6)$$

- If $i \notin \mathcal{S}$, $\hat{\beta}_i$ is set to zero, so that the zero predictors are correctly detected. That means

$$\left\| \frac{1}{N} \mathbf{X}_i^\top K \left(Y - \mathbf{X}_1 \hat{\beta_1} \right) \right\|_{\mathbb{K}} < \lambda \tilde{\omega}_i \tag{3.7}$$

To achieve a better definition of (3.6) and (3.7) we introduce the definition of Y and find, for all $i \in \mathcal{S}$

$$\|K(\beta_i^\star) - K(\hat{\beta}_i)\|_{\mathbb{K}} < \|K(\beta_i^\star)\|_{\mathbb{K}} \Rightarrow \left\| e_i^\top \left[N^{-1} \hat{\Sigma}_{11}^{-1} (\mathbf{X}_1^\top K(\varepsilon) - \lambda \tilde{s}_1) \right] \right\|_{\mathbb{K}} < \|K(\beta_i^\star)\|_{\mathbb{K}}$$

with e_i a I-size vector with all zero coefficient but the i^{th} which is 1 and $\hat{\Sigma}_{11}$ the estimated covariance matrix of $\mathbf{X}_1$: $\hat{\Sigma}_{11} = N^{-1} \mathbf{X}_1^\top \mathbf{X}_1$. While, for all $i \notin \mathcal{S}$

$$\left\| \frac{1}{N} \mathbf{X}_i^\top K \left(Y - \mathbf{X}_1 \hat{\beta}_1 \right) \right\|_{\mathbb{K}} < \lambda \tilde{\omega}_i \Rightarrow \left\| \mathbf{X}_i^\top N^{-1} \left[HK(\varepsilon) + \lambda \mathbf{X}_1 (\mathbf{X}_1^\top \mathbf{X}_1)^{-1} \tilde{s}_1 \right] \right\|_{\mathbb{K}} < \lambda \tilde{\omega}_i$$

with $H = (I - \mathbf{X}_1 (\mathbf{X}_1^\top \mathbf{X}_1)^{-1} \mathbf{X}_1^\top)$.
Considering the event $\left\{ \mathcal{S} = \hat{\mathcal{S}} \right\}$, we observe that

$$\left\{ \mathcal{S} \neq \hat{\mathcal{S}} \right\} \subset B_1 \cup B_1 \cup B_3 \cup B_4$$

with

$$\begin{aligned}
B_1 &= \left\{ \frac{1}{N} \|e_i^\top \hat{\Sigma}_{11}^{-1} \mathbf{X}_1^\top K(\varepsilon)\|_{\mathbb{K}} \geqslant \frac{\|K(\beta_i^\star)\|_{\mathbb{K}}}{2} : \text{ for some } i \in \mathcal{S} \right\} \\
B_2 &= \left\{ \frac{\lambda}{N} \|e_i^\top \hat{\Sigma}_{11}^{-1} \tilde{s}_1\|_{\mathbb{K}} \geqslant \frac{\|K(\beta_i^\star)\|_{\mathbb{K}}}{2} : \text{ for some } i \in \mathcal{S} \right\} \\
B_3 &= \left\{ \frac{1}{N} \|\mathbf{X}_i^\top HK(\varepsilon)\|_{\mathbb{K}} \geqslant \frac{\lambda \tilde{\omega}_i}{2} : \text{ for some } i \notin \mathcal{S} \right\} \\
B_4 &= \left\{ \frac{1}{N^2} \|\mathbf{X}_i^\top \mathbf{X}_1 \hat{\Sigma}_{11}^{-1} \tilde{s}_1\|_{\mathbb{K}} \geqslant \frac{\tilde{\omega}_i}{2} : \text{ for some } i \notin \mathcal{S} \right\}.
\end{aligned}$$

We will show that with N increasing $P(B_l) \to 0$ for $l = 1, \ldots 4$ and then $P(\hat{\mathcal{S}} \neq \mathcal{S}) \to 0$.

Step 1: $P(B_1) \to 0$ Given

$$B_1 = \left\{ \frac{1}{N} \|e_i^\top \hat{\Sigma}_{11}^{-1} \mathbf{X}_1^\top K(\varepsilon)\|_{\mathbb{K}} \geqslant \frac{\|K(\beta_i^\star)\|_{\mathbb{K}}}{2} : \text{ for some } i \in \mathcal{S} \right\}$$

we notice that $B_1 = \cup_{i \in \mathcal{S}} A_i$ where

$$A_i = \left\{ \frac{1}{N} \| e_i^\top \hat{\Sigma}_{11}^{-1} \mathbf{X}_1^\top K(\varepsilon) \|_{\mathbb{K}} \geqslant \frac{\| K(\beta_i^\star) \|_{\mathbb{K}}}{2} \right\} = \left\{ \frac{1}{N^2} \| e_i^\top \hat{\Sigma}_{11}^{-1} \mathbf{X}_1^\top K(\varepsilon) \|_{\mathbb{K}}^2 \geqslant \frac{\| K(\beta_i^\star) \|_{\mathbb{K}}^2}{4} \right\}$$

and we have that $P(B_1) \leqslant \sum_{i \in \mathcal{S}} P(A_i)$. For each i we have that

$$\frac{1}{N^2} \| e_i^\top \hat{\Sigma}_{11}^{-1} \mathbf{X}_1^\top K(\varepsilon) \|_{\mathbb{K}}^2 = \| K(T_i) \|_{\mathbb{K}}^2$$

where $T_i = N^{-1} e_i^\top \hat{\Sigma}_{11}^{-1} \mathbf{X}_1^\top \varepsilon$ is a Gaussian process (in $\mathbb{H}$) with zero mean and covariance operator C_T

$$\begin{aligned} C_T &= N^{-1} e_i^\top \hat{\Sigma}_{11}^{-1} \mathbf{X}_1^\top \mathbf{X}_1 \left(\hat{\Sigma}_{11}^{-1} \right)^\top e_i N^{-1} C \\ &= N^{-1} e_i^\top \hat{\Sigma}_{11}^{-1} N \hat{\Sigma}_{11} \hat{\Sigma}_{11}^{-1} e_i N^{-1} C = N^{-1} e_i^\top \hat{\Sigma}_{11}^{-1} e_i C. \end{aligned}$$

Recall C the covariance operator of the error process ε. From Corollary 1 we have that

$$P\left\{ \| K(T_i) \|_{\mathbb{K}}^2 \geqslant \theta_1 N^{-1} e_i^\top \hat{\Sigma}_{11}^{-1} e_i (\|C\|_1 + 2\|C\|_2 \sqrt{t} + 2\|C\|_\infty t) \right\} \leqslant \exp(-t).$$

Define $\tilde{t}$ such that

$$\frac{\| K(\beta_i^\star) \|_{\mathbb{K}}^2}{4} \geqslant \theta_1 N^{-1} e_i^\top \hat{\Sigma}_{11}^{-1} e_i \left(\|C\|_1 + 2\|C\|_2 \sqrt{\tilde{t}} + 2\|C\|_\infty \tilde{t} \right)$$

so then

$$\begin{aligned} P(A_i) &= P\left(\| K(T_i) \|_{\mathbb{K}} \geqslant \frac{\| K(\beta_i^\star) \|_{\mathbb{K}}}{2} \right) \\ &\leqslant P\left(\| K(T_i) \|_{\mathbb{K}}^2 \geqslant \frac{\| K(\beta_i^\star) \|_{\mathbb{K}}^2}{4} \right) \\ &\leqslant P\left(\| K(T_i) \|_{\mathbb{K}}^2 \geqslant \theta_1 N^{-1} e_i^\top \hat{\Sigma}_{11}^{-1} e_i \left(\|C\|_1 + 2\|C\|_2 \sqrt{\tilde{t}} + 2\|C\|_\infty \tilde{t} \right) \right) \\ &\leqslant \exp(-\tilde{t}) \end{aligned}$$

We can define a constant c such that

$$\left(\|C\|_1 + 2\|C\|_2 \sqrt{t} + 2\|C\|_\infty t \right) \leqslant ct$$

so that $\tilde{t}$ can satisfy the simpler inequality

$$\frac{\|K(\beta_i^\star)\|_{\mathbb{K}}^2}{4} \geqslant \frac{1}{N}\theta_1 e_i^\top \hat{\Sigma}_{11}^{-1} e_i c\tilde{t}.$$

Recall $b_N = \min_{i\in\mathcal{S}} \|K(\beta_i^\star)\|_{\mathbb{K}}$, so then

$$\frac{b_N^2}{4} \geqslant \frac{\|K(\beta_i^\star)\|_{\mathbb{K}}^2}{4} \geqslant \frac{1}{N}\theta_1 e_i^\top \hat{\Sigma}_{11}^{-1} e_i c\tilde{t}.$$

From Assumption 2.3

$$e_i^\top \hat{\Sigma}_{11}^{-1} e_i \leqslant \nu_1$$

then $\tilde{t}$ s.t.

$$\tilde{t} \leqslant \frac{Nb_N^2}{4\theta_1\nu_1 c}$$

and so, taking t equal to the upper bound we have that

$$P(A_i) \leqslant \exp\left(-\frac{Nb_N^2}{4\theta_1\nu_1 c}\right)$$

And, coming back to the statement on B_1, we can apply Assumption 2.1 to conclude that

$$P(B_1) \leqslant \sum_{i\in S} P(A_i) \leqslant I_0 \exp\left(-\frac{Nb_N^2}{4\nu_1\theta_1 c}\right) = \exp\left(-\frac{Nb_N^2}{4\theta_1\nu_1 c} + \log(I_0)\right) \to 0.$$

Step 2: $P(B_2) \to 0$ Recall that

$$B_2 = \left\{\frac{\lambda}{N}\|e_i^\top \hat{\Sigma}_{11}^{-1}\tilde{s}_1\|_{\mathbb{K}} \geqslant \frac{\|K(\beta_i^\star)\|_{\mathbb{K}}}{2} : \text{ for some } i \in \mathcal{S}\right\}$$

with $\tilde{s}_1 = \left\{N\tilde{\omega}_i\hat{\beta}_i\|\hat{\beta}_i\|_{\mathbb{K}}^{-1} \;\; i \in \mathcal{S}\right\}$. The $\mathbb{K}$ norm of $\tilde{s}_1$ is given by

$$\|\tilde{s}_1\|_{\mathbb{K}}^2 \;\; = \;\; \sum_{i\in\mathcal{S}} N^2\tilde{\omega}_i{}^2\frac{\|\hat{\beta}_i\|_{\mathbb{K}}^2}{\|\hat{\beta}_i\|_{\mathbb{K}}^2} = N^2\sum_{i\in\mathcal{S}}\tilde{\omega}_i^2 = N^2\left(\sum_{i\in\mathcal{S}}\omega_i^2 + \sum_{i\in\mathcal{S}}(\tilde{\omega}_i^2 - \omega_i^2)\right),$$

where $\tilde{w}_i = \|\tilde{\beta}_i\|_{\mathbb{H}}^{-1}$ is computed using FSL and $w_i = \|\beta_i^\star\|_{\mathbb{H}}^{-1}$. Since the $\tilde{\beta}_i$ are consistent in $\mathbb{H}$ (uniformly in i) we can apply the reverse triangle inequality several times to arrive at

$$|\tilde{\omega}_i^2 - \omega_i^2| \leqslant \frac{\|\beta_i^\star - \tilde{\beta}_i\|_{\mathbb{H}}}{\|\beta_i^\star\|_{\mathbb{H}}^3}(2 + o_P(1)),$$

where the $o_P(1)$ again holds uniformly across $i \in \mathcal{S}$. From the definition of $b_N = \min_{i\in\mathcal{S}} \|K(\beta_i^\star)\|_{\mathbb{K}}$ we have that for all $i \in \mathcal{S}$

$$b_N \leqslant \|K(\beta_i^\star)\|_{\mathbb{K}} \leqslant \theta_1^{1/2}\|\beta_i^\star\|_{\mathbb{H}}$$

and moreover from the definition of the rate r_N of Lemma (3), uniformly in i

$$\|\beta_i^\star - \tilde{\beta}_i\|_{\mathbb{H}} \leqslant \sup_{i\in\mathcal{S}}\|\beta_i^\star - \tilde{\beta}_i\|_{\mathbb{H}} = O_P(r_N^{1/2}).$$

Then, uniformly in $i \in \mathcal{S}$

$$|\tilde{\omega}_i^2 - \omega_i^2| \leqslant \frac{2}{\|\beta_i^\star\|_{\mathbb{H}}^2}\frac{\theta_1^{1/2}}{b_N}\|\beta_i^\star - \tilde{\beta}_i\|_{\mathbb{H}} \leqslant \frac{\theta_1^{1/2}}{b_N}O_P(r_N^{1/2})\omega_i^2.$$

By Assumption 2, $r_N^{1/2}/b_N \to 0$, and so we conclude

$$\begin{aligned}\|\tilde{s}_1\|_{\mathbb{K}}^2 &\leqslant N^2\left(\sum_{i\in\mathcal{S}}\omega_i^2\right)(1 + o_p(1)) = N^2\left(\sum_{i\in\mathcal{S}}\frac{1}{\|\beta_i^\star\|_{\mathbb{H}}^2}\right)(1 + o_p(1)) \\ &\leqslant N^2\frac{I_0\theta_1^2}{b_N^2}(1 + o_p(1)). \qquad (3.8)\end{aligned}$$

Then for the original object we have for each $i \in \mathcal{S}$

$$\frac{\lambda}{N}\frac{\|e_i^\top\hat{\Sigma}_{11}^{-1}\tilde{s}_1\|_{\mathbb{K}}}{\|K(\beta_i^\star)\|_{\mathbb{K}}} \leqslant \frac{\lambda}{N}\frac{\|e_i^\top\hat{\Sigma}_{11}^{-1}\|\,\|\tilde{s}_1\|_{\mathbb{K}}}{\|K(\beta_i^\star)\|_{\mathbb{K}}}$$

with $\|e_i^\top\hat{\Sigma}_{11}^{-1}\| \leqslant \|e_i\|\|\hat{\Sigma}_{11}^{-1}\|_{op} \leqslant \nu_1$ form Assumption 2 and in the end

$$\frac{\lambda}{N}\frac{\|e_i^\top\hat{\Sigma}_{11}^{-1}\|\,\|\tilde{s}_1\|_{\mathbb{K}}}{\|K(\beta_i^\star)\|_{\mathbb{K}}} \leqslant \frac{\lambda\nu_1\sqrt{I_0}N}{Nb_Nb_N}(1 + o_p(1)) \to 0.$$

Step 3 From the previous definition of B_3:

$$B_3 = \left\{ \frac{1}{N} \|\mathbf{X}_i^\top HK(\varepsilon)\|_{\mathbb{K}} \geqslant \frac{\lambda \tilde{\omega}_i}{2} : \text{ for some } i \notin \mathcal{S} \right\}$$

we define A_i s.t. for $i \notin \mathcal{S}$

$$A_i = \left\{ \frac{1}{N} \|\mathbf{X}_i^\top HK(\varepsilon)\|_{\mathbb{K}} \geqslant \frac{\lambda \tilde{\omega}_i}{2} \right\}$$

and $B_3 = \cup_{i \notin \mathcal{S}} A_i$. We can define the gaussian process $\mathbf{X}_i H\varepsilon$, which has zero mean and as covariance operator $\mathbf{X}_i^\top HH^\top \mathbf{X}_i C = \mathbf{X}_i^\top H\mathbf{X}_i C$, since H is symmetric and idempotent, with C the covariance operator of the zero mean gaussian process ε. Moreover, since we have that $\sup_{i \notin \mathcal{S}} \|\tilde{\beta}_i\|_{\mathbb{H}} = O_P\left(r_N^{1/2}\right)$ we can notice that $\tilde{\omega}_i \leqslant 1/\sup_{i \notin \mathcal{S}}(\|\tilde{\beta}_i\|_{\mathbb{H}})$ and then

$$A_i \subseteq \left\{ O_P\left(r_N^{1/2}\right) \|\mathbf{X}_i^\top HK(\varepsilon)\|_{\mathbb{K}} \geqslant \frac{N\lambda}{2} \right\}.$$

Then for any $\epsilon > 0$ we can find a $T = T(\epsilon) > 0$ s.t.

$$P(A_i) \leqslant \frac{\epsilon}{2(I - I_0)} + P\left(\|\mathbf{X}_i^\top HK(\varepsilon)\|_{\mathbb{K}} \geqslant \frac{N\lambda}{2Tr_N^{1/2}} \right).$$

As we discussed before, to apply Corollary 1, we need to detect $\tilde{t}$ s.t.

$$\mathbf{X}_i^\top H\mathbf{X}_i(\|C\|_1 + 2\|C\|_2\sqrt{\tilde{t}} + 2\|C\|_\infty \tilde{t}) \leqslant \left(\frac{N\lambda}{2Tr_N^{1/2}} \right)^2. \tag{3.9}$$

Focusing on the left side of the inequality we know that

$$\mathbf{X}_i^\top H\mathbf{X}_i(\|C\|_1 + 2\|C\|_2\sqrt{\tilde{t}} + 2\|C\|_\infty \tilde{t}) \leqslant N\tilde{t}c.$$

Since H is a projection matrix we have

$$\mathbf{X}_i H\mathbf{X}_i = \sum_{t=1}^{N} \left(\sum_{n=1}^{N} \mathbf{X}_{i,n} H_{n,t} \right)^2 = \sum_{t=1}^{N} 1 = N,$$

and again there exists a constant c such that $\forall t, ct \geqslant (\|C\|_1 + 2\|C\|_2\sqrt{t} + 2\|C\|_\infty t)$, so we define $\tilde{t}$:

$$\tilde{t}cN \leqslant \left(\frac{N\lambda}{2Tr_N^{1/2}}\right)^2 \Rightarrow \tilde{t} = \frac{\lambda^2 N}{4T^2 c r_N}.$$

Applying corollary 1 we have

$$P\left(\|X_i^\top HK(\varepsilon)\|_{\mathbb{K}} \geqslant \frac{N\lambda}{2Tr_N^{1/2}}\right) \leqslant \exp\left(-\frac{\lambda^2 N}{4T^2 c r_N}\right) \leqslant \exp\left(-\frac{I_0 \log^2(I)}{N4T^2 c r_N}\right)$$

and then we can compute the probability of B_3

$$\begin{aligned} P(B_3) &\leqslant \sum_{i \notin S} P(A_i) \leqslant (I - I_0)\exp\left(-\frac{I_0 \log^2(I)}{4NT^2 c r_N}\right) + \frac{\epsilon}{2} \\ &\leqslant \exp\left(-\frac{I_0 \log^2(I)}{4NT^2 c r_N} + \log(I - I_0)\right) + \frac{\epsilon}{2}. \end{aligned}$$

Since $r_N \ll (I_0 \log^2(I))/N$, we can take N large enough to make the first term smaller then $\epsilon/2$ and have the convergence of the probability to 0.

Step 4 Recall that B_4 is defined as

$$B_4 = \left\{\frac{1}{N^2}\|X_i^\top X_1 \hat{\Sigma}_{11}^{-1} \tilde{s}_1\|_{\mathbb{K}} \geqslant \frac{\tilde{\omega}_i}{2} : \text{ for some } i \notin S\right\}.$$

Recall from (3.8)

$$\|\tilde{s}_1\|_{\mathbb{K}}^2 \leqslant N^2 \theta_1^2 \frac{I_0}{b_N^2}(1 + o_p(1)),$$

as well as

$$\sup_{i \notin S} \tilde{\omega_i}^{-1} = O_P(r_N^{1/2}).$$

The irrepresentable condition implies

$$\forall i \notin S, \ \|X_i^\top X_1 \hat{\Sigma}_{11}^{-1}\|_{op} \leqslant \|\hat{\Sigma}_{21}\hat{\Sigma}_{11}^{-1}\|_{op} \leqslant \phi < 1.$$

Then we consider the inequality of B_4 for a fixed $i \notin S$

$$\frac{2\|\mathbf{X}_i^\top \mathbf{X}_1 \hat{\Sigma}_{11}^{-1} \tilde{s}_1\|_{\mathbb{K}}}{N^2 \tilde{\omega}_i} \leqslant \frac{2\|\mathbf{X}_i^\top \mathbf{X}_1 \hat{\Sigma}_{11}^{-1}\|_{op} \|\tilde{s}_1\|_{\mathbb{K}}}{N^2 \tilde{\omega}_i} \leqslant \frac{2\phi r_N^{1/2} I_0^{1/2} \theta_1}{N b_N} O_P(1) \to 0,$$

which finishes Step 4 and completes the proof.

3.5.3 *The week oracle property: Theorem 1.2*

Let $h_n = \{h_{i,n}\} \in \mathbb{K}^I$ be a bounded sequence: $\|h_n\|_{\mathbb{K}} < M_1$. We will show that

$$\frac{\sqrt{N}\langle h_n, \hat{\beta} - \beta^\star \rangle_{\mathbb{H}}}{\sigma_n} \xrightarrow{D} \mathcal{N}(0,1) \qquad \text{where} \qquad \sigma_n^2 = \sum_{i=1}^{I_0} \sum_{i=1}^{I_0} \hat{\Sigma}_{11;ij}^{-1} \langle h_{i,n}, C h_{j,n} \rangle,$$

assuming that the $h_{i,n}$ are chosen such that $\sum_{i \in S} \langle C^{1/2} h_i, C^{1/2} h_i \rangle \geqslant M_2 > 0$ for some fixed M_2. Recall that the oracle estimator is

$$\hat{\beta}_O^S = (\mathbf{X}_1^\top \mathbf{X}_1)^{-1} \mathbf{X}_1^\top Y \qquad \text{and} \qquad \hat{\beta}_O = \{\hat{\beta}_O^S, 0\},$$

where 0 here is the zero function in $\mathbb{K}^{I-I_0}$. Since we assume that the Y are Gaussian, we have that

$$\sqrt{N}\langle h_n, \hat{\beta}_O - \beta_1^\star \rangle_{\mathbb{H}} \sim \mathcal{N}(0, \sigma_n^2).$$

By Assumption 2.3 we have that

$$\sigma_n^2 \geqslant \nu_1^{-1} \sum_{i \in S} \langle C^{1/2} h_i, C^{1/2} h_i \rangle \geqslant \nu_1 M_2,$$

and so is bounded from below, so we need only to show that

$$\sqrt{N}\langle h_n, \hat{\beta}_O - \hat{\beta}_1 \rangle_{\mathbb{H}} = o_P(1).$$

From equation 3.5, when $\hat{S} = S$ we have that

$$\begin{aligned}\sqrt{N}\langle h, \hat{\beta}_O - \hat{\beta}\rangle_{\mathbb{H}} &= \sqrt{N}\lambda\langle (\mathbf{X}_1^\top \mathbf{X}_1)^{-1} K^{-1}(\tilde{s}_1), h_n^S\rangle_{\mathbb{H}} \\ &= \frac{\lambda}{\sqrt{N}}\langle \Sigma_{11}^{-1} K^{-1/2}(\tilde{s}_1), K^{-1/2} h_n^S\rangle \\ &\leqslant \frac{\lambda}{\sqrt{N}} \|\Sigma_{11}^{-1} K^{-1/2}(\tilde{s}_1)\|_{\mathbb{H}} \|h_n\|_{\mathbb{K}}.\end{aligned}$$

Applying Assumption 2.3 we have that

$$\frac{\lambda}{\sqrt{N}} \|\Sigma_{11}^{-1} K^{-1/2}(\tilde{s}_1)\|_{\mathbb{H}} \|h_n\|_{\mathbb{K}} \leqslant \frac{\lambda}{\sqrt{N}\nu_1} \|\tilde{s}_1\|_{\mathbb{K}} \|h_n\|_{\mathbb{K}}.$$

From the equation (3.8) we have

$$\|\tilde{s}_1\|_{\mathbb{K}} \leqslant \frac{\sqrt{I_0} N}{b_N}(1 + o_p(1))$$

and then

$$|\sqrt{N}\langle h, \hat{\beta}_O - \hat{\beta}_1\rangle_{\mathbb{H}}| \leqslant \frac{\lambda\sqrt{I_0}\sqrt{N}\|h_n\|_{\mathbb{K}}}{\nu_1 b_N}(1 + o_P(1)) = o_P(1),$$

by Assumption 2. Since $P\left(\hat{S} = S\right) \to 1$ the proof is complete.

3.5.4 *The strong oracle property: Theorem 2*

We begin by partitioning the problem into two pieces:

$$\begin{aligned}N\|\hat{\beta} - \hat{\beta}_O\|^2 &= N\sum_{i=1}^{I} \|\hat{\beta}_i - \hat{\beta}_{O;i}\|^2 \\ &= N\sum_{i=1}^{I}\sum_{i=1}^{J} \langle \hat{\beta} - \hat{\beta}_O, e_i \otimes \nu_j\rangle^2 \qquad (3.10) \\ &+ N\sum_{i=1}^{I}\sum_{i=J+1}^{\infty} \langle \hat{\beta} - \hat{\beta}_O, e_i \otimes \nu_j\rangle^2. \qquad (3.11)\end{aligned}$$

Bounding (3.10) follows the similar arguments as in the proof of 1.2, namely

$$\langle \hat{\beta} - \hat{\beta}_O, e_i \otimes v_j \rangle^2 = \frac{\lambda^2}{N^2 \theta_j} \langle \hat{\Sigma}_{11}^{-1} K^{-1/2}(\tilde{s}_1), e_i \otimes v_j \rangle^2 \leqslant \frac{\lambda^2}{N^2 \theta_j \nu_1^2} \langle K^{-1/2}(\tilde{s}_1), e_i \otimes v_j \rangle^2.$$

This gives the bound

$$N \sum_{i=1}^{I} \sum_{i=1}^{J} \langle \hat{\beta} - \hat{\beta}_O, v_j \rangle^2 \leqslant \frac{\lambda^2}{\theta_J \nu_1^2 N} \|\tilde{s}_1\|_{\mathbb{K}}^2 \leqslant \frac{\lambda^2 N I_0}{\theta_J \nu_1^2 b_N^2} (1 + o_P(1)).$$

Turning to the second term, we express $\hat{\beta}$ using a different form. Notice that we can write

$$\tilde{s}_1 = \Lambda \hat{\beta}_1,$$

where Λ is a diagonal matrix of the terms $\{N\tilde{w}_i \|\hat{\beta}_i\|_K^{-1}\}$. We therefore have that

$$\mathbf{X}_1^\top K(Y) - (\mathbf{X}_1^\top \mathbf{X}_1) K(\hat{\beta}) - \lambda \Lambda \hat{\beta}_1 = 0$$

We can re-express this equation as

$$\hat{\beta}_O - \hat{\beta}_1 + \lambda (\mathbf{X}_1^\top \mathbf{X}_1)^{-1} \Lambda K^{-1}(\hat{\beta}_1) = 0 \Rightarrow \hat{\beta}_1 = (I + \lambda (\mathbf{X}_1^\top \mathbf{X}_1)^{-1} \Lambda K^{-1})^{-1} \hat{\beta}_O.$$

The above shrinks (all operators above are positive definite) every coordinate of $\hat{\beta}_O$ to obtain $\hat{\beta}_1$ and thus we have that

$$N \sum_{j=J+1}^{\infty} \sum_{i=1}^{\infty} \langle \hat{\beta} - \hat{\beta}_O, e_i \otimes v_j \rangle^2 \leqslant 4N \sum_{j=J+1}^{\infty} \sum_{i=1}^{\infty} \langle \hat{\beta}_O, e_i \otimes v_j \rangle^2.$$

We compute the expected value

$$E\langle \hat{\beta}_O, e_i \otimes v_j \rangle^2 = \langle \beta^\star, e_i \otimes v_j \rangle^2 + (\mathbf{X}_1^\top \mathbf{X}_1)_{i,i}^{-1} \langle C v_j, v_j \rangle.$$

This implies that

$$\begin{aligned} 4N \sum_{j=J+1}^{\infty} \sum_{i=1}^{\infty} \langle \hat{\beta}_O, e_i \otimes v_j \rangle^2 = \\ = \quad O_P(1) N \left[\sum_{i=1}^{I} \sum_{j=J+1}^{\infty} \langle \beta^\star, e_i \otimes v_j \rangle^2 + \sum_{i=1}^{I} \sum_{j=J+1}^{\infty} (\mathbf{X}_1^\top \mathbf{X}_1)_{i,i}^{-1} \langle C v_j, v_j \rangle \right]. \end{aligned}$$

Which can be bounded by

$$O_P(1)\left[NI_0\theta_J^{1+\delta}B^2 + \frac{I_0}{\nu_1}o(1)\right],$$

as long as $J \to \infty$, since C is a trace class operator.

To ensure both (3.10) and (3.11) go to zero, we require that J is such that

$$N\theta_J^{1+\delta} \to 0 \qquad \text{and} \qquad \frac{\lambda^2 N}{\theta_J b_N^2} \to 0.$$

So we need to be able to choose J such that

$$\theta_J \ll N^{-1/(1+\delta)} \quad \text{and} \quad \theta_J \gg \frac{\lambda^2 N}{b_N^2}.$$

This is possible if

$$\frac{\lambda^2 N}{b_N^2} \ll N^{-1/(1+\delta)} \Leftrightarrow \lambda \ll \frac{b_N}{N^{1/2[1+1/(1+\delta)]}},$$

as desired.

4

MICROBIOME AND GROWTH CRURVES: STOOL AND BUCCAL MICROBIOMES INFLUENCE GROWTH CURVES OF CHILDREN

4.1 INTRODUCTION

Due to the increasing number of young children already meeting criteria of overweight, as presented by Ogden et al. (2014), many studies have been recently carried on to analyze the obesity risk for children. In particular, several works highlight strong connections between early childhood weight gain and increased later risk of overweight (Fisch et al. (1975) and Whitaker et al. (1997)), but no early prevention interventions have been conducted (Hesketh and Campbell (2010)). The Intervention Nurses Start Infants Growing on Healthy Trajectories (INSIGHT) of Paul et al. (2014) is a controlled trial to evaluate the effectiveness of early intervention to prevent rapid infant weight gain. The study follows families from the early pregnancy at least up to the end of the second year of age of the first born child. Parents are taught how to maintain a responsive feeding and healthy education. Children are followed in their first 3 years and monitored by nurses four times in their first year of age and annually by the experts of the clinical research center. Data collected during this follow up are divided into different categories: "Anthropometrics and Biological Specimens", "Child behavior", "Parenting", "Maternal Psychosocial Variables and behavior", "Family Context" and "Background, Demographics and Covariates". The analysis we present in this chapter focus on the "Anthropometrics and Biological Specimens" subset of data and in particular on the periodic measurements of weight and height of children, on their stool and buccal microbiome and on the saliva microbiome of the mother. Several studies have already been presented to identify connections between microbiome and overweight conditions (Ley et al. (2007), Kalliomäki et al. (2008), Koleva et al. (2015)); but

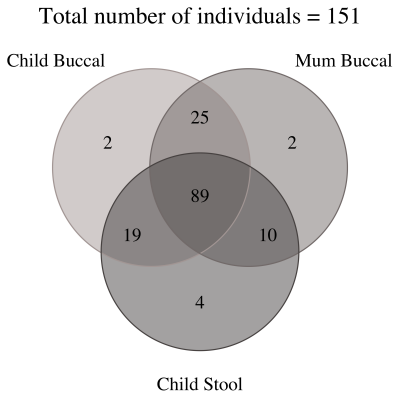


FIGURE 4.1: Venn Diagram of the microbiome data collected for the 151 children sample

the novelty of our approach is the consideration of the longitudinal measure of the growth of children combined with the analysis of the oral microbiome.

This chapter is organized as follow. In Section 4.2 we present the methods used to preprocess growth curves, as well as the microbiome data, while in Section 4.3 we introduce apply functional linear models to these data and finally in Section 4.4 preliminary results and further developments are presented.

4.2 PROCESSING OF DATA

In this Section we present the dataset we use. For each of the $n = 1, \dots, N = 151$ children we collect measurements on their growth in the first 2 or 3 years of age and some microbiome data (collected during the 2 years of age visit). Regarding the growth, we have 6 measurements for the height and the weight of children in their first 2 years of age and for some of them ($\sim$ 20%) an additional measurement at around 3 years of age. Giving that we are considering young children, the BMI is not a suitable measure to analyze their growth. Thus we consider as response $y_n(t)$ the ratio between weight [kg] and height [m], as proposed by the W.H.O. (1948). Focusing on the microbiome, for each child we collect the complete microbiome at least for one of the following samples: the stool of the child (Child Stool sample), the saliva of the child (Child Buccal sample) and the saliva of the mother (Mother Buccal sample). The three microbiome sets are collected at the Genus level, a taxonomic rank above Species and below Family. Given that not all the children have all the data for the three microbiomes, as we

can see from Figure 4.1, and since the three microbiomes have intrinsically different structures, we decide to take the analysis separately for the three sets.

In the next sections we focus on the preliminary analysis on the growth measurements (Section 4.2.1) and on the microbiome (Section 4.2.2 and 4.2.3) to present the data we consider in the further regression analyses.

4.2.1 *The growth curves*

A classical approach (Ramsay and Silverman (2005)) to define the function $y(t)$ from a set of observations (t_j, y_j) is the representation of $y(t)$ as a linear combination of a known (and well defined) set of functions $s_1(t), \dots, s_P(t)$, called basis:

$$y(t) = \sum_{p=1}^{P} \gamma_p s_p(t),$$

where P is the dimension of the basis and γ_p are the coefficients of the linear combination. The most common choice of basis for non-periodic data (as in our case of height and weight measurements) is the spline basis, since it guarantees a sufficient regularity level, maintaining easy expression and computational efficiency. Classically we focus on the B-spline basis, introduced by Boor (2001) which has the advantage to be well conditioned and with an explicit recursive formula (Cox de Boor formula of Boor (2001)).

In the particular case of the longitudinal measurement of the growth index, the setting is very sparse (7 measurements in the first 3 years of age) and many patients don't even have the measurements for the last year/years of follow up. Thus, we decide to tighten the grid of the measurements and extend the domain of the functions $y_n(t)$ with a Functional Principal Component approach, proposed by Chen et al. (2016) and Yao et al. (2005). This approach is very efficient to take advantage of the total set of data aiming to infer the missing points and tighten the originally sparse grid of measurements. Moreover, given that less than 20% of children have a measurement at their 3 years of age, we focus only in the first two years of follow up. A further analysis could be the selection of patients with measurements on the all 3 years to inspect whether the microbiome changes its influence in the third year of life of the child. To conclude, once we have selected the measurements of the weight/height ratio in the first two years of life, we can define the growth curves as functions of the L^2 space considering the following approach:

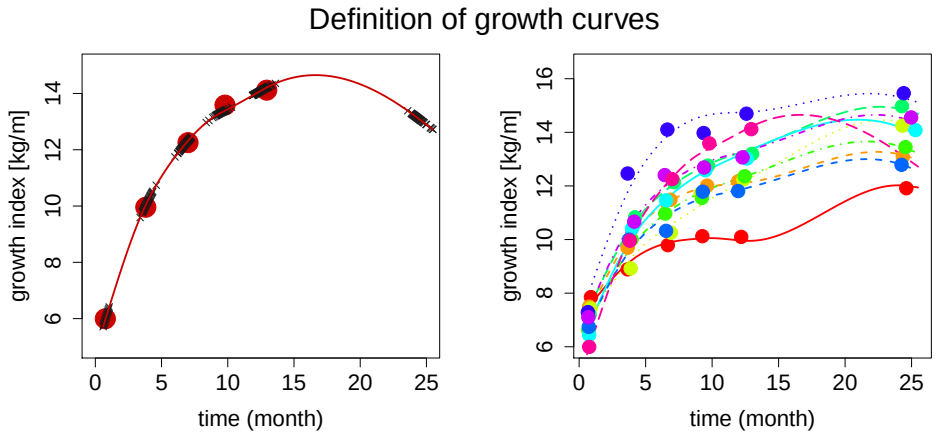


FIGURE 4.2: Smoothing of the growth curves. Left panel: an example of smoothing for a patient, red points are the original measurements, black crosses the tightened set estimated via FPCA and the solid line is the final smoothed function. Right panel: a set of 10 patients, the points are the original measurements and the solid lines are the final curves estimated.

1. apply the Functional Principal Component approach described in Chen et al. (2016) and Yao et al. (2005) (and implemented in the `refund` R package of Huang et al. (2016)) to tighten the grid, estimate points out of the domain and then improve the smoothing we are going to perform with splines.

2. apply to this new tightened set of observations the spline approximation with the definition of the cubic B-spline basis with evenly spaced knots.

An example of this procedure is shown in the left panel of Figure 4.2 where the red dark points are the observed values of a randomly selected child, the black crosses are the points introduced with the FPCA approach (in step 1.) and the solid line is the final spline approximation. In the right panel, instead, the original measurements of 10 children are plotted and the correspondent final splines are drawn.

Finally, the smoothed curves are registered to isolate and remove the phase variability. The registration is performed with continuous warping functions $h_n \in \mathcal{W}$, where $\mathcal{W}$ is chosen as the set of spline functions represented by a 10 elements cubic B-spline basis with evenly knots. Moreover, to keep the time domain invariant, the h_n are bounded at the extremes: $h_n(0) = 0$ and $h_n(T) = T \ \ \forall n$. Specifically, here we align the derivatives of

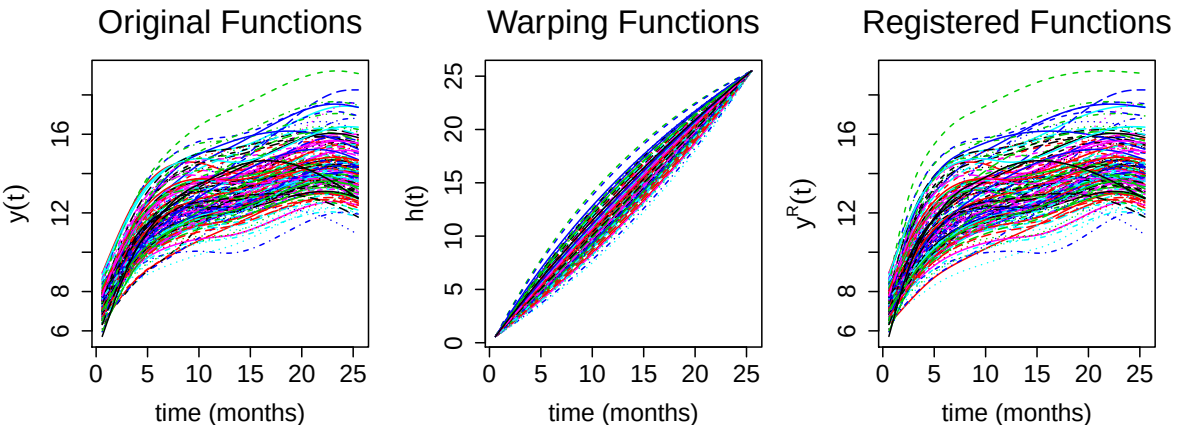


FIGURE 4.3: Registration procedure. Left panel: original growth curves. Central panel: Warping functions estimated to minimize the distance between the registered derivatives and the mean derivative. Right panel: final set of registered curves.

the curves and then compute the registered functions. In Figure 4.3 the original functions are presented together with the optimal warping functions and the resulting registered curves.

4.2.2 *The α-diversities of the microbiome data*

For each of the three sets of microbiome, we collected the measurements of the abundances of bacteria in the sample. These data are collected at the Genus level and they consist of $I = 1,055$ bacteria. As proposed by Heltshe and Forrester (1983), the global composition of the microbiome of each child can play a key role in the microbiome analysis. This global composition should be analyzed taking into account both the richness of the microbiome, i.e. the number of different Genera present, and the eveness of the microbiome, i.e. the relative abundance of each Genus. The α-diversity measurements are a quantification of this global structure and have been recently connected to overweight condition. Specifically, diversity of the microbiome seems to be lower in obese individuals when compared to normal-weight individuals (see for example Ursell et al. (2012)). To quantify the α-diversity we here consider the Inverse Simpson Index, as proposed by Simpson (1949). Given a set of T abundances $s_1, \dots, s_T$ at the Genus level, saying Q the total

number of abundances $Q = \sum_{t=1}^{T} s_i$, the Simpson Index, known also as the Hunter-Gaston index, is computed as

$$\mathcal{S} = \frac{\sum_{t=1}^{T} t_i(1-t_i)}{Q(Q-1)};$$

this index takes into account both the richness and the evenness of the abundances. $\mathcal{S}$ takes small values in datasets of high diversity, while it assumes large values in datasets of low diversities and this counterintuitive characteristics leads to the introduction of new α-diversity indices. These are obtained as transformations of $\mathcal{S}$, but are characterized by an increasing value, following the higher diversity. For example, we can introduce the Inverse Simpson Index $1/\mathcal{S}$, or the Gini-Simpson index $1-\mathcal{S}$. In the further analyses we will focus on the Inverse Simpson Index

$$\alpha\text{-diversity} = \frac{1}{\mathcal{S}}$$

that has 1 as lowest value and the higher the index is, the greater the diversity is.

4.2.3 *The microbiome abundances*

Beside considering the global composition of the abundances of the Inverse Simpson Index, we can also present analyses on the whole set of Genus level abundances. Before considering them in the next regression steps, we present a new pipeline to treat this original set, merging the abundances to remove their sparse and correlated structure. This procedure aims to identify and treat the sparsity structure of the abundances (*thicking procedure*) and their correlations (*liaising procedure*). These two steps are carried on taking into account the Phylogenetic structure of the Genera collected.

The analysis here is referred to the Child Buccal sample, but the same pipeline has been applied unchanged also to the other two samples (details are presented in Supplementary Figures: Supplementary Figure S4.3, S4.4 for the Child Stool sample and Supplementary Figure S4.5, S4.6 for the Mother Buccal sample).

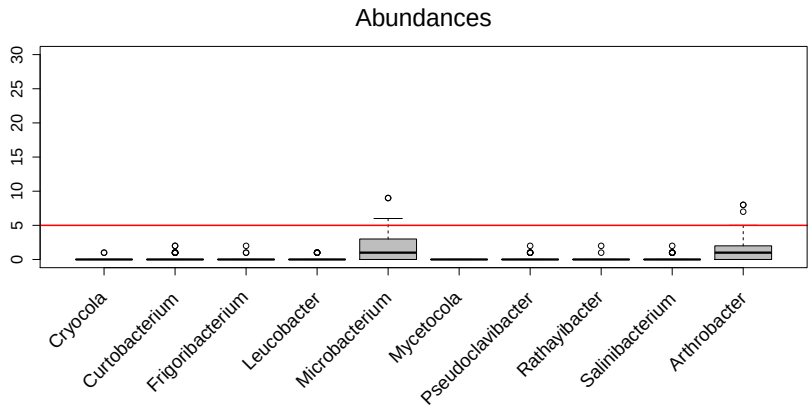


FIGURE 4.4: Boxpolot of 10 abundances for the Child Buccal sample. The red line is the threshold imposed for the definition of negligible abundances. In this plot only `Microbacterium` is considered as relevant. All the other samples are present in a negligible proportion of data.

The thicking procedure

From a visual inspection of data, we notice that many of the collected abundances have negligible level (close to 0 for many children); thus forcing us to treat these data separately. We consider a level as negligible if the abundance assumes a small value (lower than a fixed threshold -e.g. 5-) for more than 90% of the samples. In Figure 4.4 an example of the distribution of 10 abundances in the Child Buccal sample is shown. In this subset just the *Microbacterium* is considered as relevant and kept for the following analysis. In Figure 4.5 the Phylogeny tree of the $1,055$ bacteria is shown; red points indicate the negligible abundances, while green points are the non negligible ones. The proportion of negligible abundances is very high ~ 0.89 (~ 0.87 for the Child Stool sample and ~ 0.88 for the Mother Buccal); then a method to remove this sparsity structure and then *thick* the sample is necessary.

Specifically, considering the Phylogenic informations we have on the genera of the bacteria, we group the non significant abundances following their Phylogenetic structure: the abundances are summed together considering the Phylogenetic relationship. Basically, the merging procedure consists of two steps: the first influences only the deeper level of the tree, trying to merge, i.e. sum, all possible brothers without affecting the other levels of the Phylogeny; the second step affects also the deeper levels allowing the possibility of merging bacteria even if their are connected with branches of length two or three or don't belong to the deeper level of the tree.

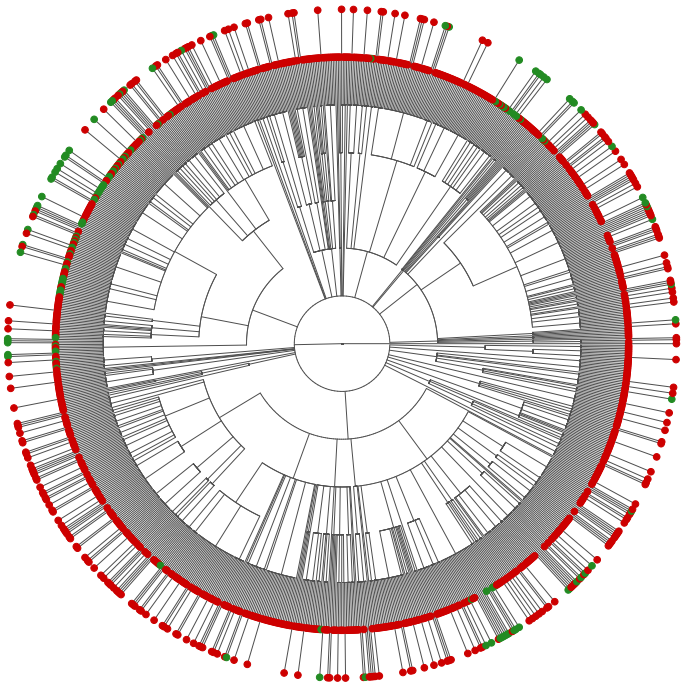


FIGURE 4.5: Phylogeny of the bacteria recorded for the Child Buccal sample. Red points are the bacteria to be neglected and green to be kept. As we can notice from the prevalence of red points, only 11% of abundances have non negligible value.

In Figure 4.6 an example of the *thicking procedure* on a subtree extracted from the original Phylogeny with the Child Buccal abundances is shown. In Table 4.1 a detail of the number of bacteria (or sets of bacteria) present in the three samples after the *thicking procedure* is presented.

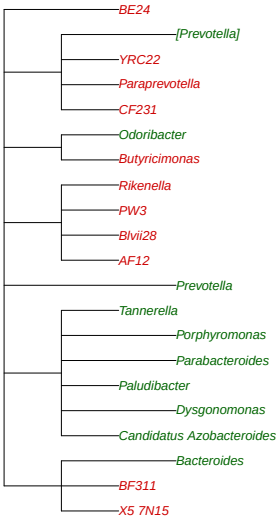
The liaising procedure

Once we have the tree correspondent to the *thicken* abundances, we need to consider their correlation structure. Computing their pairwise correlation we notice that some of the abundances have high correlation (yellow points of Supplementary Figure S4.1 have correlation higher than 0.7) and these abundances are mainly close in the Phylogeny. Then, we consider to average the highly correlated abundances which are also close in the Phylogeny. We propose an iterative procedure that follows the Phylogenetic structure of the abundances and whether it detects an high correlated pair of abundances, it merges them computing their average. As for the previous merging in Table 4.1, the size of the samples after the *liaising procedure* is shown. To maintain the possibility of comparing the abundances, after each averaging step abundances are standardized. In Supplementary Figure S4.2 the final correlation plot is shown; the number of high correlated abundances is significantly decreased, even if some high correlated pairs are still present, but not directly connected in the Phylogeny. Computing the minimum and maximum eigenvalues σ of the correlation matrices obtained, we get $\sigma_{min} = 5.3\,10^{-5}$ and $\sigma_{max} = 17.29$ for the Child Buccal sample, $\sigma_{min} = 1.3\,10^{-4}$ and $\sigma_{max} = 11.86$ for the Child Stool sample and $\sigma_{min} = 1.1\,10^{-4}$ and $\sigma_{max} = 16.69$ for the Mother Buccal sample. These results allow this new dataset to be used as set of predictors for the next regression analyses.

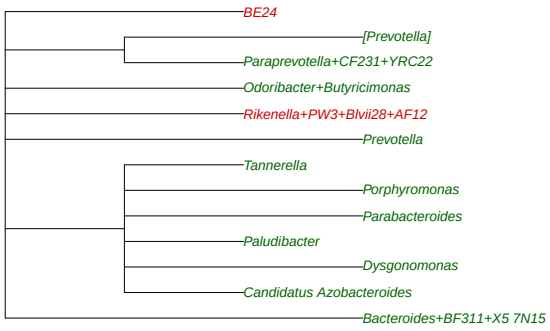
Finally, in Figure 4.7 the final tree for the Child Buccal sample after the *thicking* and *liaising procedure* is shown and in Annex Table 1 the ID codification of the abundances or of the groups obtained after the merging procedure is presented.

4.3 THE INFLUENCE OF MICROBIOME ON THE GROWTH OF CHILDREN

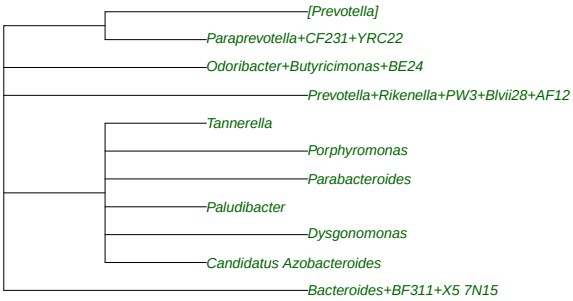
In the next sections we present two separate analyses on the influence of the microbiome on the growth curves. Specifically, we refer to the influence of the α-diversities (Section 4.3.1), but also to the influence of the whole set of merged Genus level abundances (Section 4.3.2) proposing some regression models.



(A) Original subtree.



(B) Subtree after the first merging step.



(C) Subtree after the second merging step.

FIGURE 4.6: *thicking procedure* of the subtree of panel (A). Red names correspond to negligible abundances in the Child Buccal sample; green names to non negligible. In panels (B) and (C) the two *thicking* steps are presented and abundances summed are named adding a + to connect the two original abundances. At the end of the *thicking procedure* all the abundances have non negligible level, as shown by the green names in panel (C).

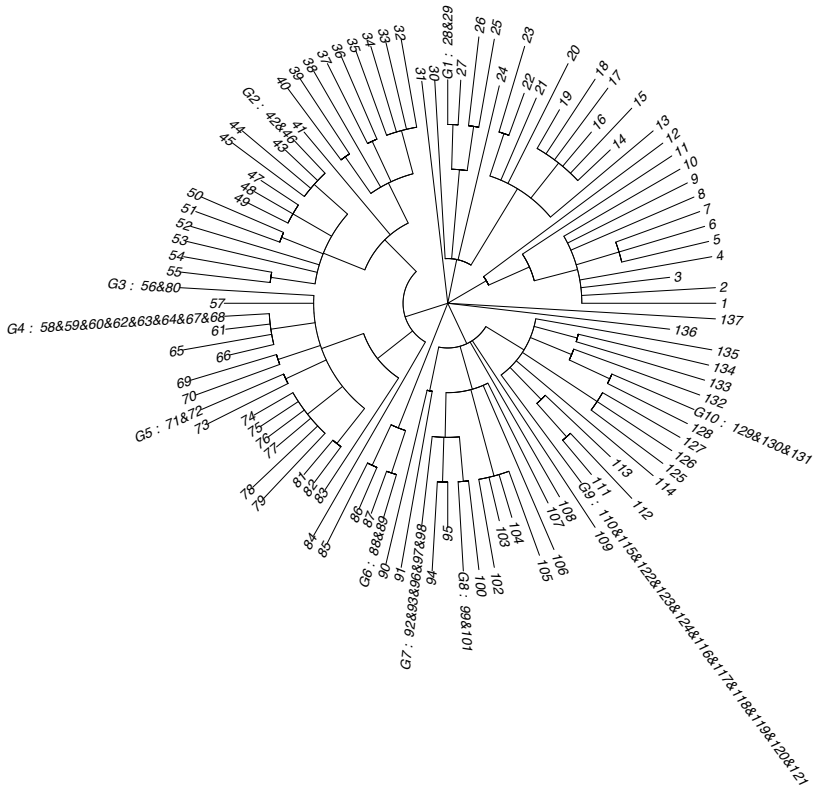


FIGURE 4.7: Phylogenetic tree for the Child Buccal sample, after the application of the *thicking* and *liaising procedure*. Names and group IDs are listed in Annex Table 1.

	Sample		
	Child Buccal	Child Stool	Mother Buccal
Initial non negligible Abundances	118	134	131
After *thicking procedure*	137	149	147
After *liaising procedure*	106	103	108

TABLE 4.1: Number of abundances in the three samples. Specifically the initial number of non negligible abundances, the number of abundances after the *thicking procedure* and the final number after the *liaising procedure* are shown.

4.3.1 *The influence of α-diversities on the growth curves*

As a preliminary analysis we consider children at 2 years of age and we classify them in terms of growth index as below or above average. We focus on this specific time point since it is when the microbiome samples have been collected. We aim to identify whether the α-diversities are connected to the growth condition at this specific point. In Figure 4.8 we present the box plot of the α-diversities dividing children as above and below average at 2 years.

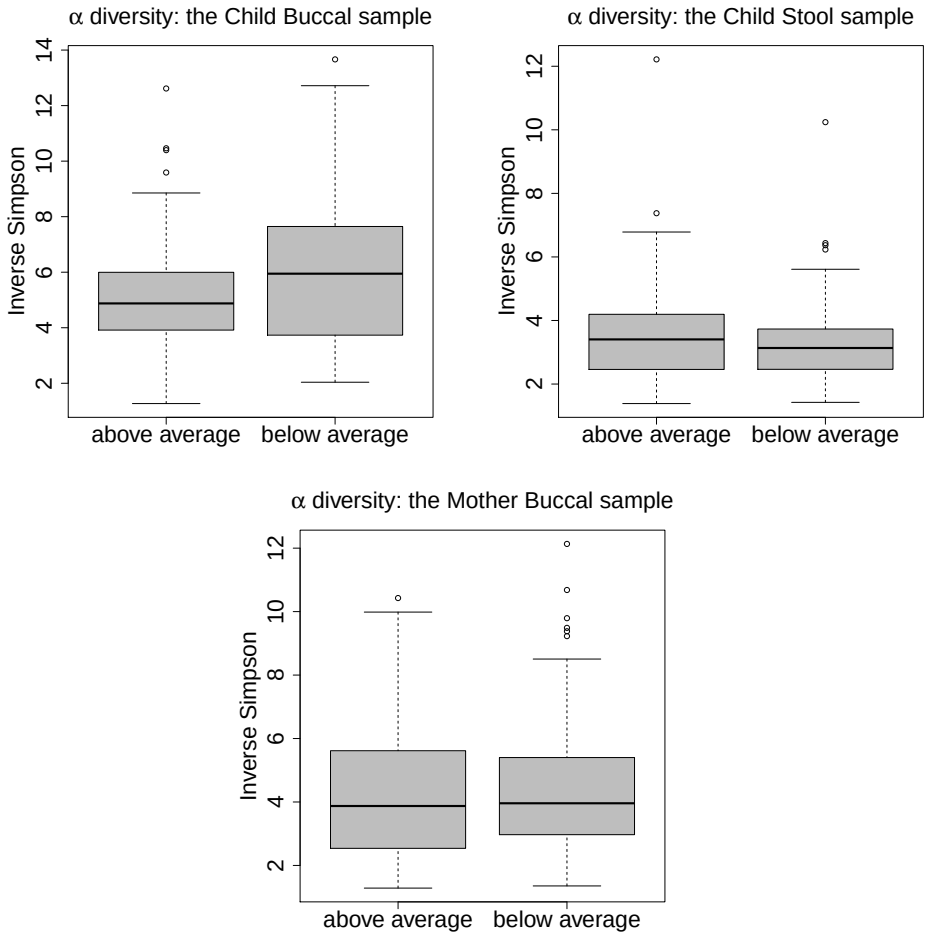


FIGURE 4.8: Boxplot of the α-diversities in above and below average 2 years aged children for the Child Buccal, Child Stool and Mother Buccal dataset.

Testing the differences in the averages of the two populations (two-sided z-tests) we obtain high significance to claim that the average of the Child Buccal α-diversity is different in the two populations (p-value 0.0136), while no evidence confirms this difference for the Child Stool and Mother Buccal sample (p-values 0.2827 and 0.5904). Given the evidence for the influence of the α-diversities to the growth of children at this specific time point, we now aim to inspect their time-dependent influence. We don't focus any more on the measurements of the growth index at 2 years of age, but on their longitudinal dimension. We apply the Function-on-Linear regression estimation proposed by Ramsay and Silverman (2005), with the penalty term set to 0, to quantify the time-dependent effect of the α-diversities on growth curves. In Figure 4.9 we present the estimation of the regression coefficients for the three separate analyses. The dotted curves indicate plus and minus 2 estimated standard errors for the estimation. There is high evidence to confirm that the α-diversity of the Child Buccal dataset is related to the growth of children: an increase of the diversity reflects a growth index below average during the whole time domain. Moreover, computing on this sample the p-value to test the significancy of the estimated coefficients with the L^2 norm-test presented by Horváth and Kokoszka (2012), we obtain 0.005. This confirms the connection between the Child Buccal α-diversity and the growth curves of children. Regarding the Child Stool and Mother Buccal sample, instead, the two bands include the value 0 for all time domain, and the p-values are respectively, 0.73 and 0.28; then we cannot prove the connection between growth and this two samples of α-diversities. However, the trends confirm the visual inspection of the box-plots, with a positive effect of the Child Stool α-diversity, i.e. an increasing diversity is connected to above average children, and a negative effect of Mother Buccal α-diversity, i.e. an increasing diversity is connected to below average children.

4.3.2 *The influence of abundances on the growth curves*

In this section we apply the FLAME method introduced in Chapter 3 to isolate the influence of the Genus level abundances on the growth curves. The three microbiomes are analyzed separately and the three regression results are presented in Figure 4.11.

Detailing the procedure, in Figure 4.10 the growth curves are presented in their original registered version and once the point-wise mean has been subtracted; we apply regression to de-trended curves to detect bacteria which cause an increment/decrement of the growth index of children with respect

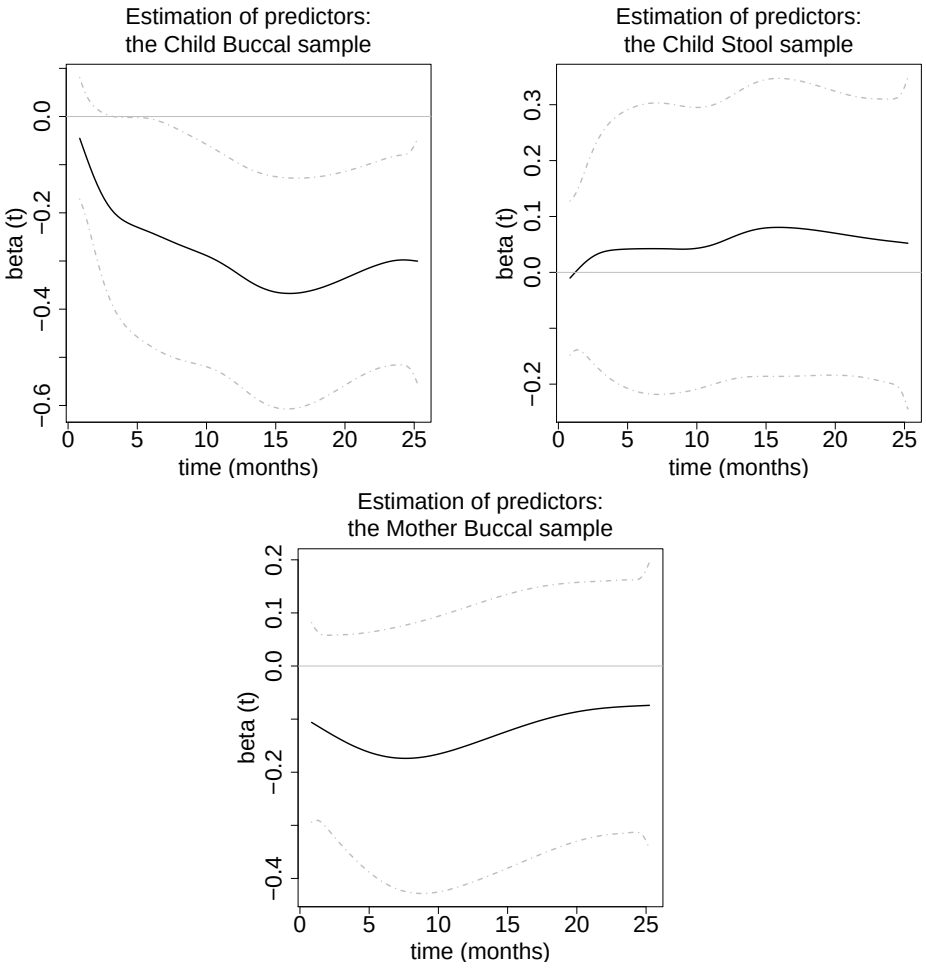


FIGURE 4.9: Estimation of the functional coefficients associated to the three α-diversity samples performed as presented by Ramsay and Silverman (2005). Bands are computed pointwise as mean ± 2 standard deviation. To improve the visual representation of the result, plots are associated to an estimation with the smoothing regression parameter $\lambda = 1$.

to the average. FLAME is run with the Sobolev kernel and the smoothing parameter $\sigma = 8$. The grid for λ is a 200 points grid (or 500 for the Child Buccal microbiome) going from the λ_{max} that sets all the coefficients to zero, as it is defined in the original method, up to $r_\lambda \cdot \lambda_{max}$, with $r_\lambda = 0.001$. The plot of the final regression coefficients estimated by FLAME for the three microbiome samples are shown in Figure 4.11.

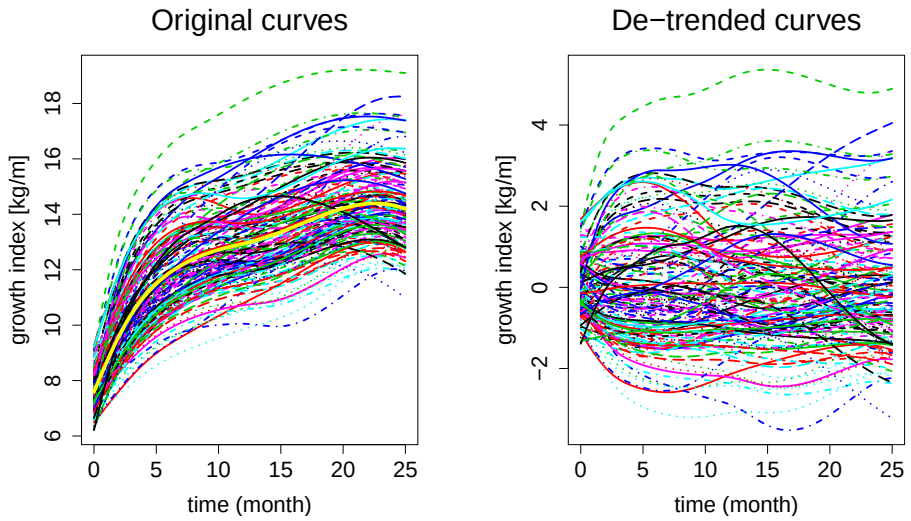


FIGURE 4.10: Growth curves. Left panel: original data after the smoothing procedure of Section 4.2.1. Right panel: residual from the point-wise mean.

To validate the regression analysis we compute a R^2 index to highlight the proportion of variability explained by the model:

$$R^2 = 1 - \frac{\sum_{n=1}^{N} \|y_n - \hat{y}_n\|_{L^2}^2}{\sum_{n=1}^{N} \|y_n - \bar{y}_N\|_{L^2}^2}$$

where $y_1, \ldots, y_N \in \mathbb{H}$ are the growth curves, $\hat{y}_1, \ldots, \hat{y}_N \in \mathbb{H}$ are the curves fitted with FLAME and $\bar{y}_N$ is the empirical mean. We obtain $R^2 = 0.11$ for the Child Buccal sample; $R^2 = 0.32$ for the Child Stool sample and $R^2 = 0.08$ for the Mother Buccal sample and these results confirm the possibility of explaining a relevant part of the variability of the growth curves with the microbiome. To compute an unbiased version of the R^2 (R^2_{cv}), i.e. not affected by the use of the same set of data for both the estimation and the computation of the error, we divided the dataset into two balanced partitions and used one part as *training set* to estimate the model and the other as *test set*. Defining the first half of the dataset as the *training set* and the second half as the *test set*, we get for the Child Stool sample $R^2_{cv} = 0.055$ and $R^2_{cv} = 0.013$ for the Child Buccal sample. Regarding the Mother Buccal sample, instead, the introduction of the predictors does not improve the estimation obtained with the simple average in the cross-validated setting. This preliminary result confirms that both the Stool and the Buccal microbiome

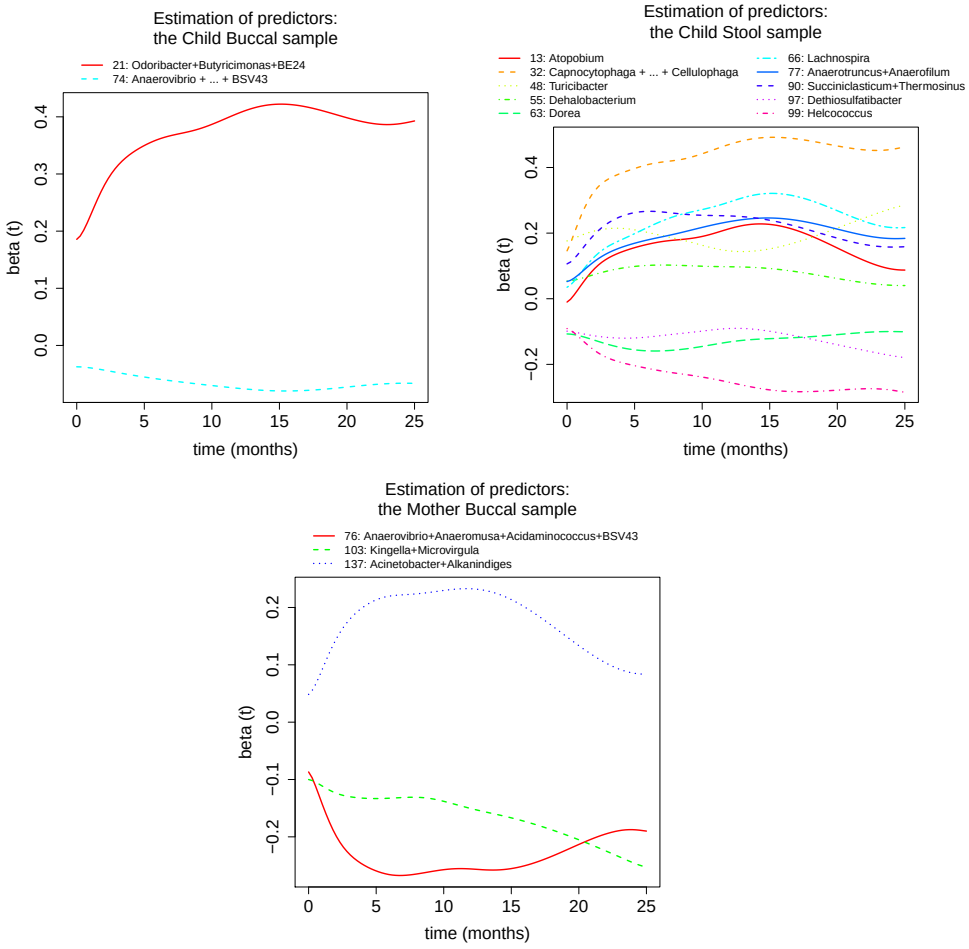


FIGURE 4.11: FLAME estimation for the Child Buccal, Child Stool and Mother Buccal dataset. Only significant coefficients detected by FLAME are plotted.

of the Child, if analyzed as the global set of abundances at the Genus level, influence the growth curves of children.

4.4 DISCUSSION

Focusing on the Child Stool microbiome, many articles have already highlighted the relationship between the composition of the microbiome and the predisposition to obesity. Turnbaugh et al. (2006), for example, focus on the Phylum Bacteroidetes and the Phylum Firmicutes showing how a vari-

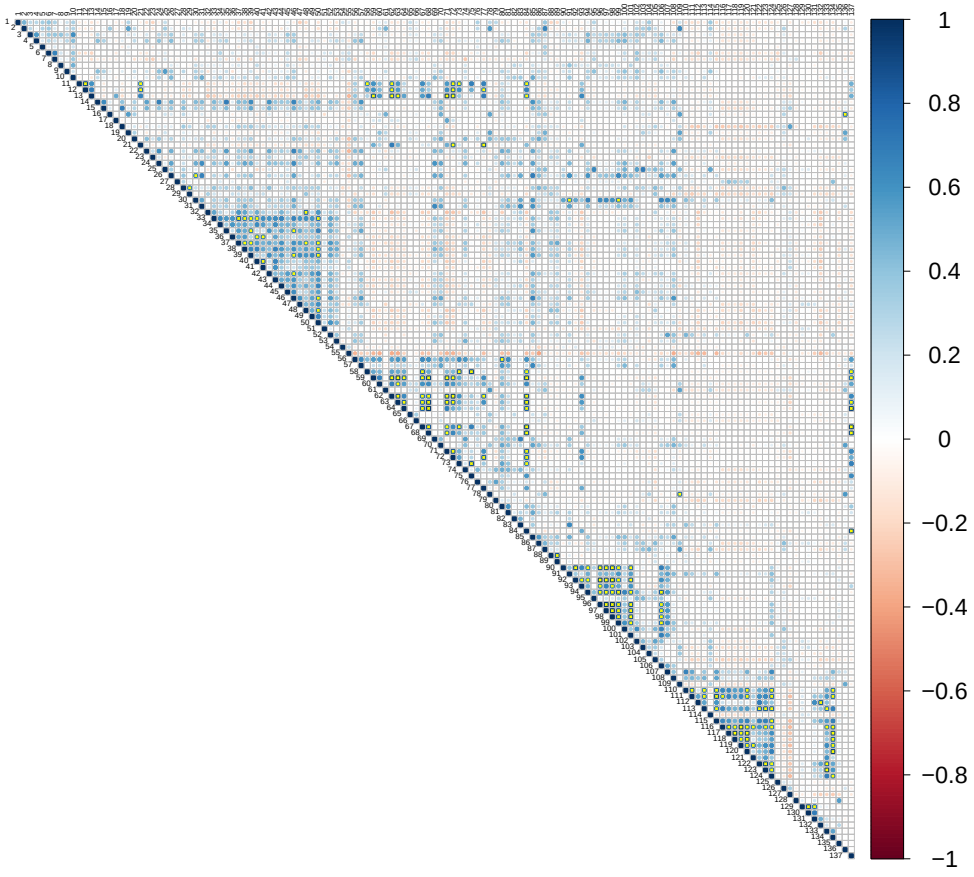
ation of the abundances of these Phyla in the stool microbiome of adults can be related to obesity. Specifically, they conclude that an increasing abundance of the Firmicutes bacteria, related to a decreasing abundance of the Bacteroidetes Phylum, can be related to obesity. However, the intestinal microbiome of the first years of age of children is very different from the one of the adults and few studies focused on these data (Arrieta et al. (2014)). Moreover, the study we are introducing here is focused on the abundances at the Genus level, not at the Phylum (or at Class) level as in the main paper on the effects of the microbiome on the phenotypical expression of individuals.

In this preliminary analysis we detect a relationship between the microbiome and the overweight condition of children: the presence above the average of a set of 5 bacteria of the Flavobacteriales Class of the Bacteroidetes is related to a ratio weight/height above average. Recent studies have confirmed that the presence of the Phylum of Bacteroidetes in the microbiome stool of children can be related to the high fiber intake that maximizes the metabolic energy extraction from ingested plant polysaccharides. Focusing on the Firmicutes, instead, literature highlight that the increasing level of this Phylum of bacteria in adults is related to an increasing obesity level. In the analysis we carry on here we notice that in children some Genera are related to an increase of the ratio weight/height, but some of them (Dorea, Dethiosulfatibacter and Helcococus) are related to a weight ratio under the average. Finally, the Atopobium Genus, beloging to the Actinobacteria Phylum, is linked to an obesity level greater than average. This bacteria, like the Bacteroidetes Phylum, is related to the digestion of fibers and polysaccharides.

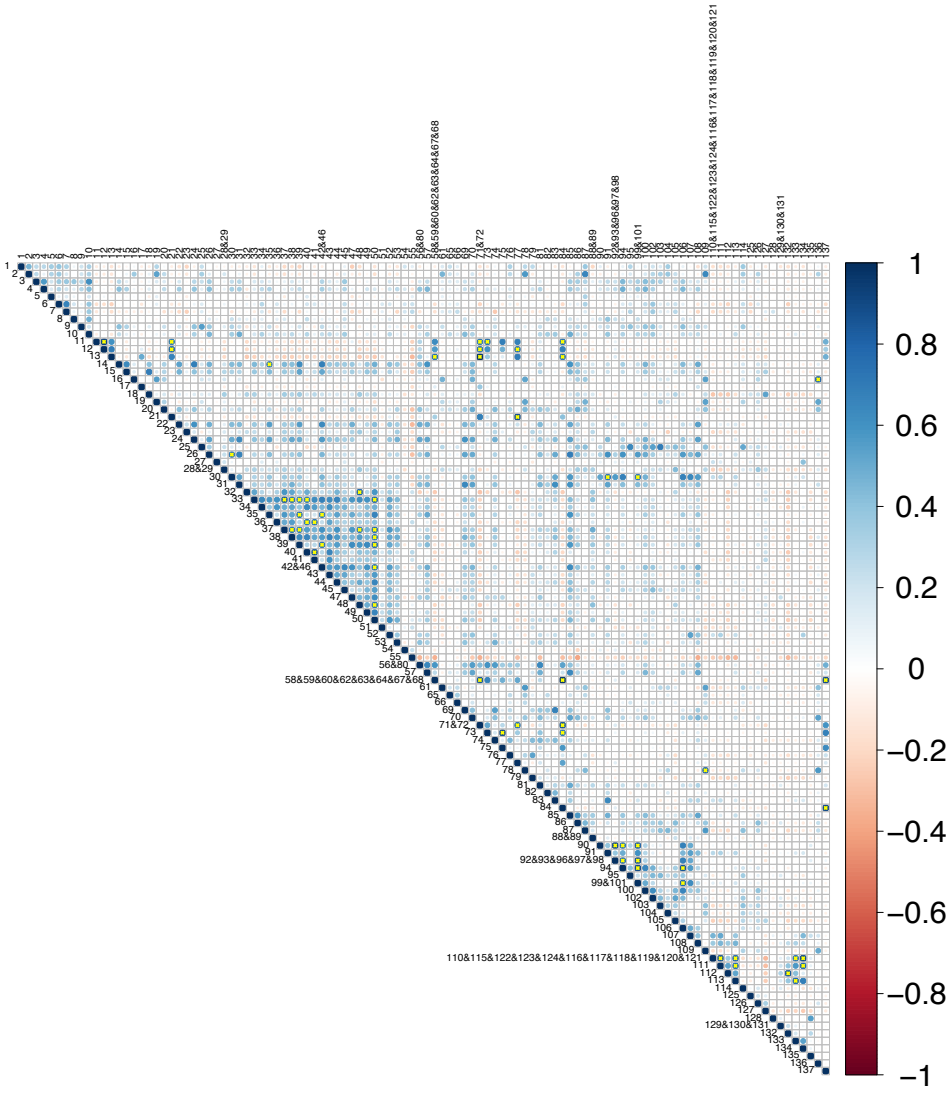
In this work we analyze also the oral microbiome of both child and mum, trying to detect the set of bacteria in the saliva microbiome which influence the growth of children. As we presented before, in literature many analyses have been conducted to detect the influence of the gut microbiome in the growth of children, but very few tried to link the composition of the oral microbiome to the growth of individuals. In this chapter we report some results to define the connection between oral microbiome and the growth index of children in their first years. A first relevant aspect is detected from the α-diversity analysis: we notice that an increasing diversity in the microbiome is connected to a under average growth index throughout the whole time domain of interest, and significantly affects the growth of children in the second year of life. A second aspect derives from the analyses of the influence of the Genus abundances: the preliminary result we have presented shows that both in Mother and Child microbiome, a presence above average

of the Firmicutes Phylum group *Anaerovibrio + Anaeromusa + Acidaminococcus + BSV43* is connected to a reduction of the growth index throughout the considered time domain. On the contrary, regarding the Child microbiome, the presence of the Odornobacter Genus, belonging to the Bacteroidetes Phylum, produces an increment in the growth index.

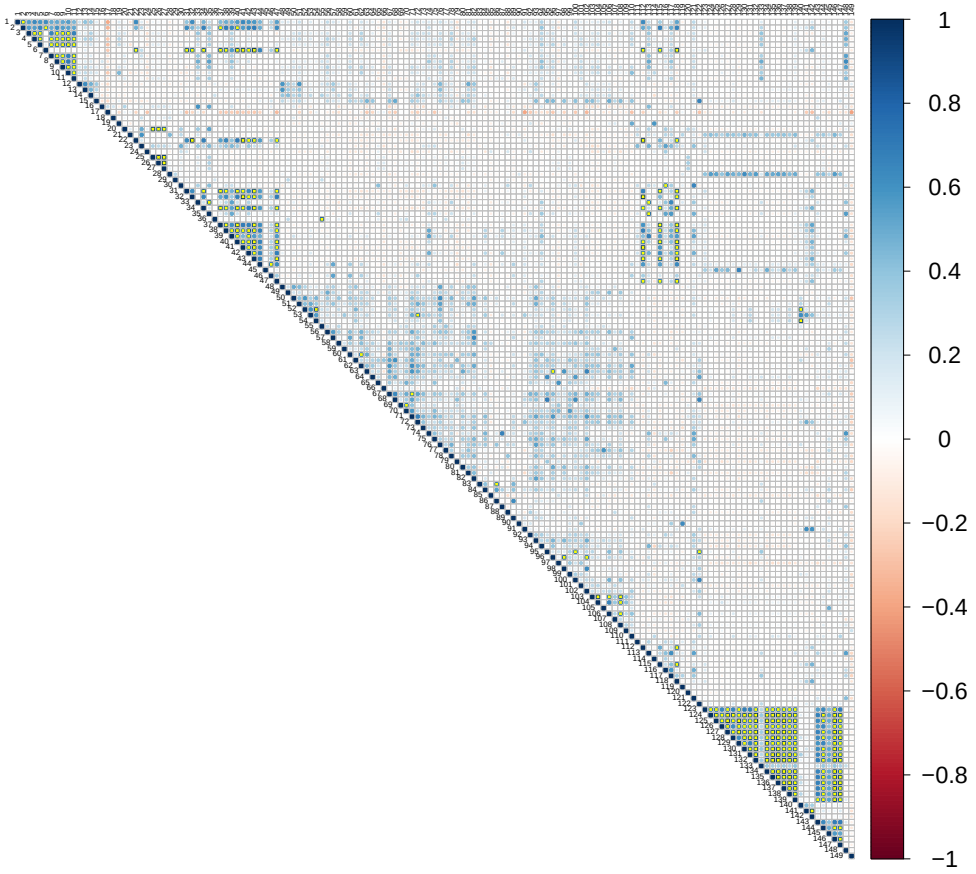
The analyses we have presented in this chapter are the first preliminary results from the INSIGHT study. Nevertheless, we still have to deep-dive on this in order to include many further details, as for example the diet informations we have collected during the follow-up of patients or the antibiotic exposure in early life. Moreover, we are planning to analyze metabolomics data from the Nuclear Magnetic Resonance spectroscopy analysis (Bernstein et al. (1957)) of the Stool sample to enrich our analysis.



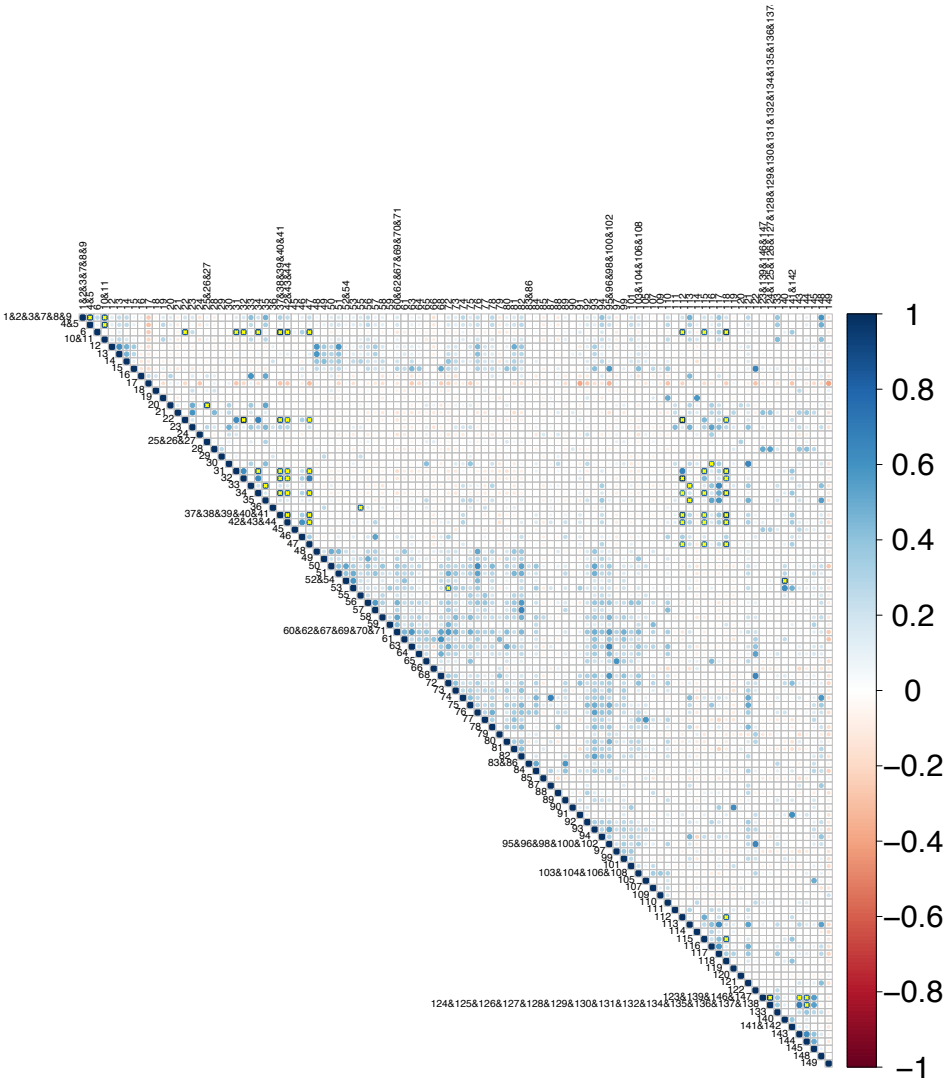
SUPPLEMENTARY FIGURE S4.1: Original correlation structure for the abundances of the Child Buccal sample. Names of the abundances are omitted and syntetized with numberic IDs; for the correspondance of the IDs with the names see Annex Table 1. We can notice that some abundances close in the Phylogeny have correlation higher than the threshold and then a merging procedure is needed to make the linear regression model applicable.



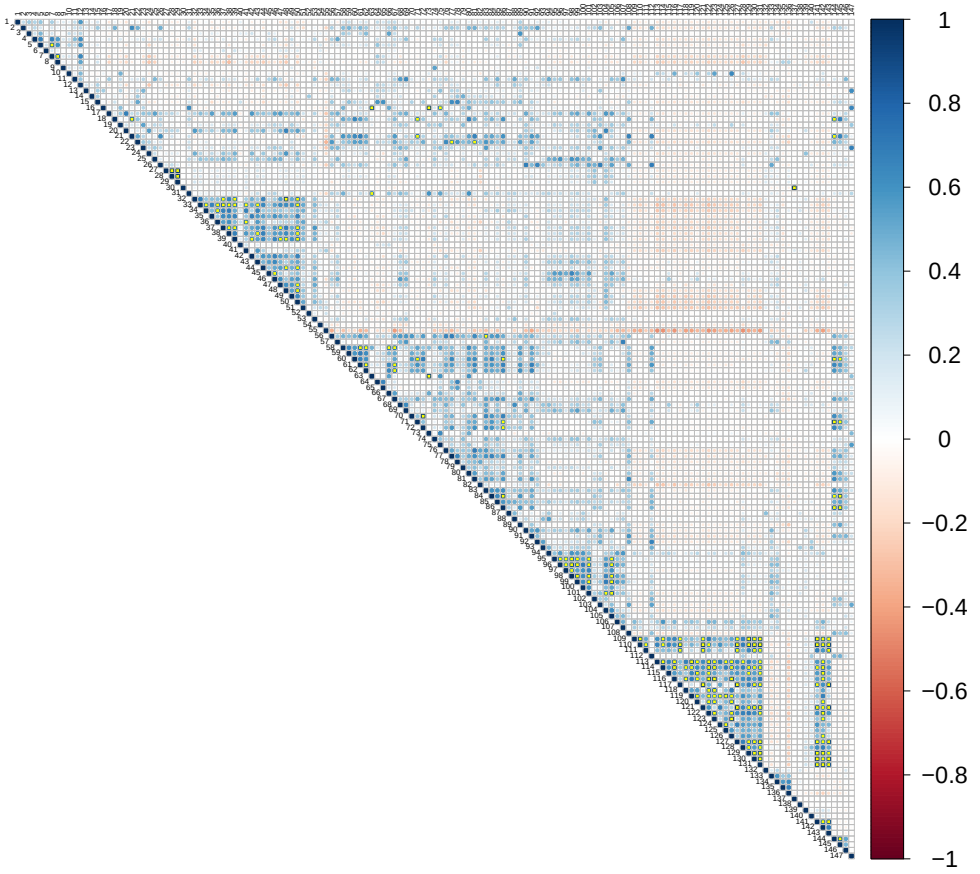
Supplementary Figure S4.2: Final correlation structure for the abundances. Averaged samples are named with the original IDs connected by &. After the merging procedure the majority of the abundances with correlations above the threshold have been removed. Some still have correlation above 0.7, but cannot be merged since they are not directly connected on the tree.



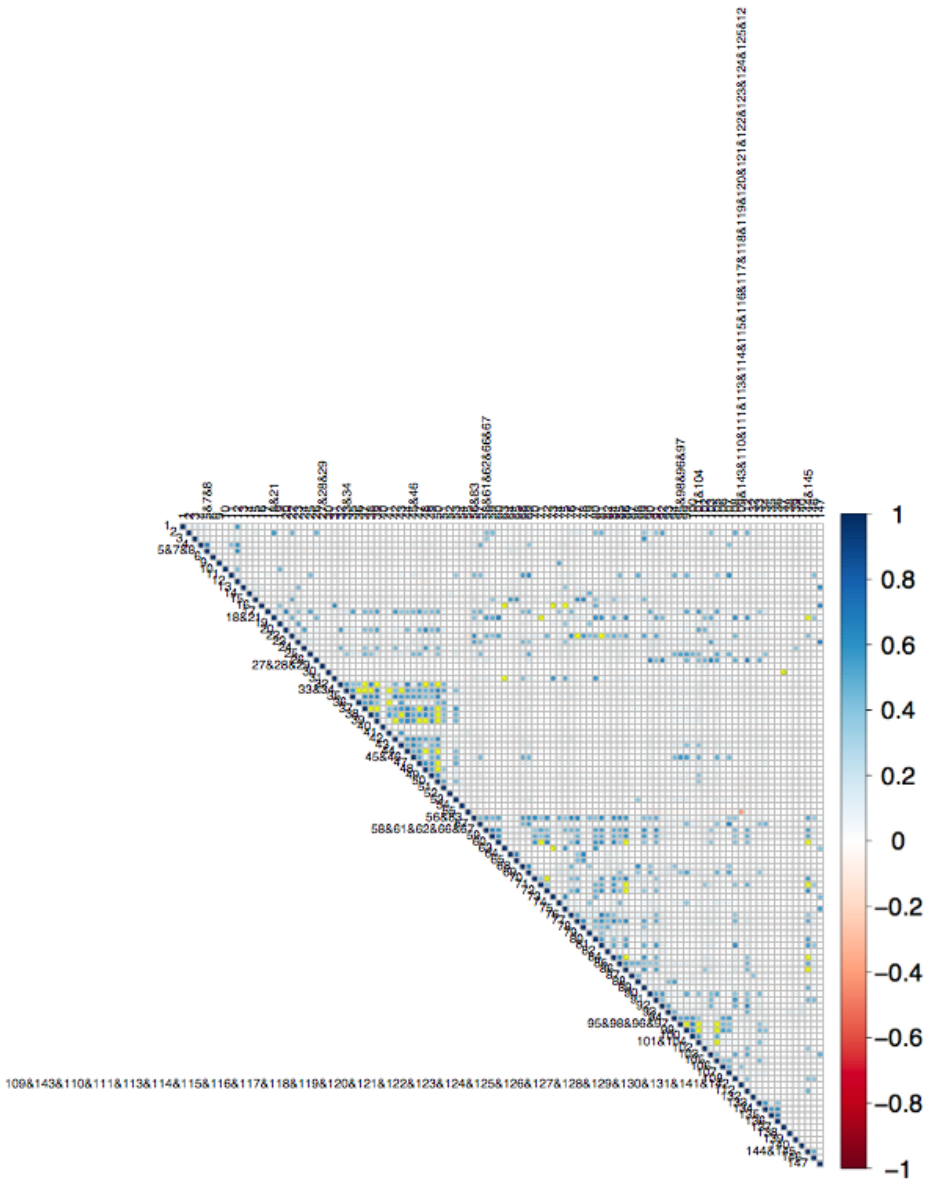
Supplementary Figure S 4.3: Original correlation structure for the abundances of the Child Stool sample. Names of the abundances are omitted and synthesized with numerical IDs; for the correspondence of the IDs with the names see Annex Table 2. Yellow points represent the correlations above the threshold of 0.7.



Supplementary Figure S 4.4: Final correlation structure for the abundances of the Child Stool sample. Averaged samples are named with the original IDs connected by &. As for the Child Buccal sample there are still some abundances with correlation higher than 0.7, but are not merges since they are not close in the Phylogeny.



Supplementary Figure S 4.5: Original correlation structure for the abundances of the Mother Buccal sample. Names of the abundances are omitted and synthesize with numerical IDs; for the correspondence of the IDs with the names see Annex Table 3. As for the previous cases yellow points represent high correlated abundaces.



Supplementary Figure S 4.6: Final correlation structure for the abundances of the Mother Buccal sample. Averaged samples are named with the original IDs connected by &. Still there are some high correlated abundances which are not merged since they are not Phylogenetically directly linked.

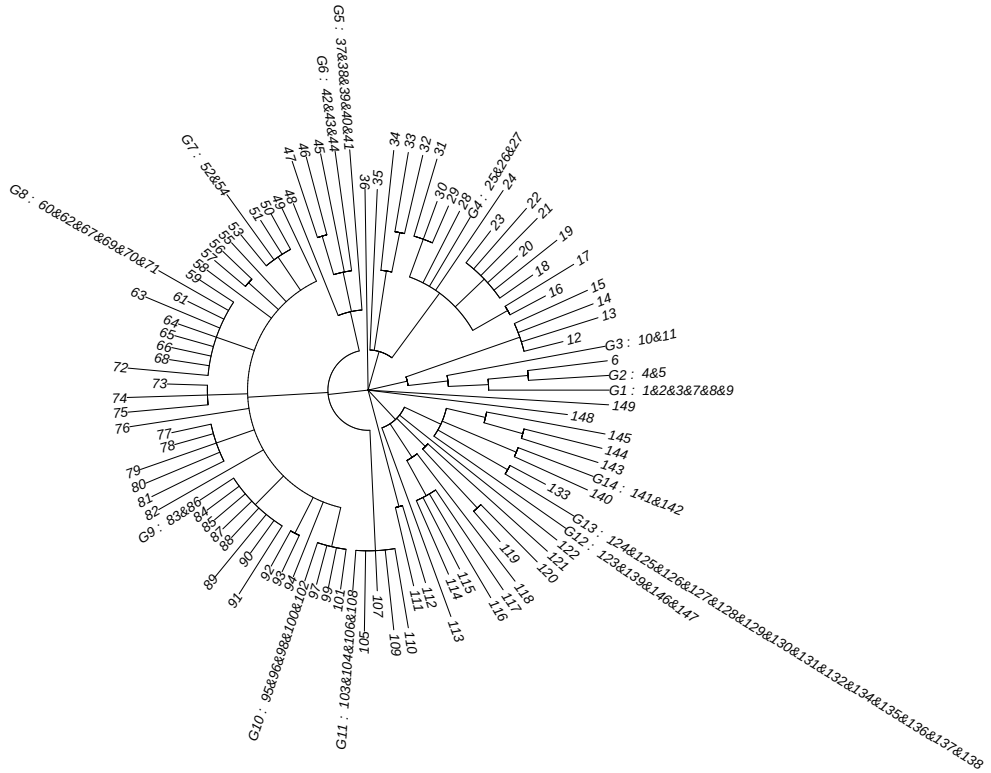


FIGURE 4.7: Phylogenetic tree for the Child Stool sample, after the application of the *thicking* and *liaising procedure*. Names and group IDs are listed in Annex Table 2.

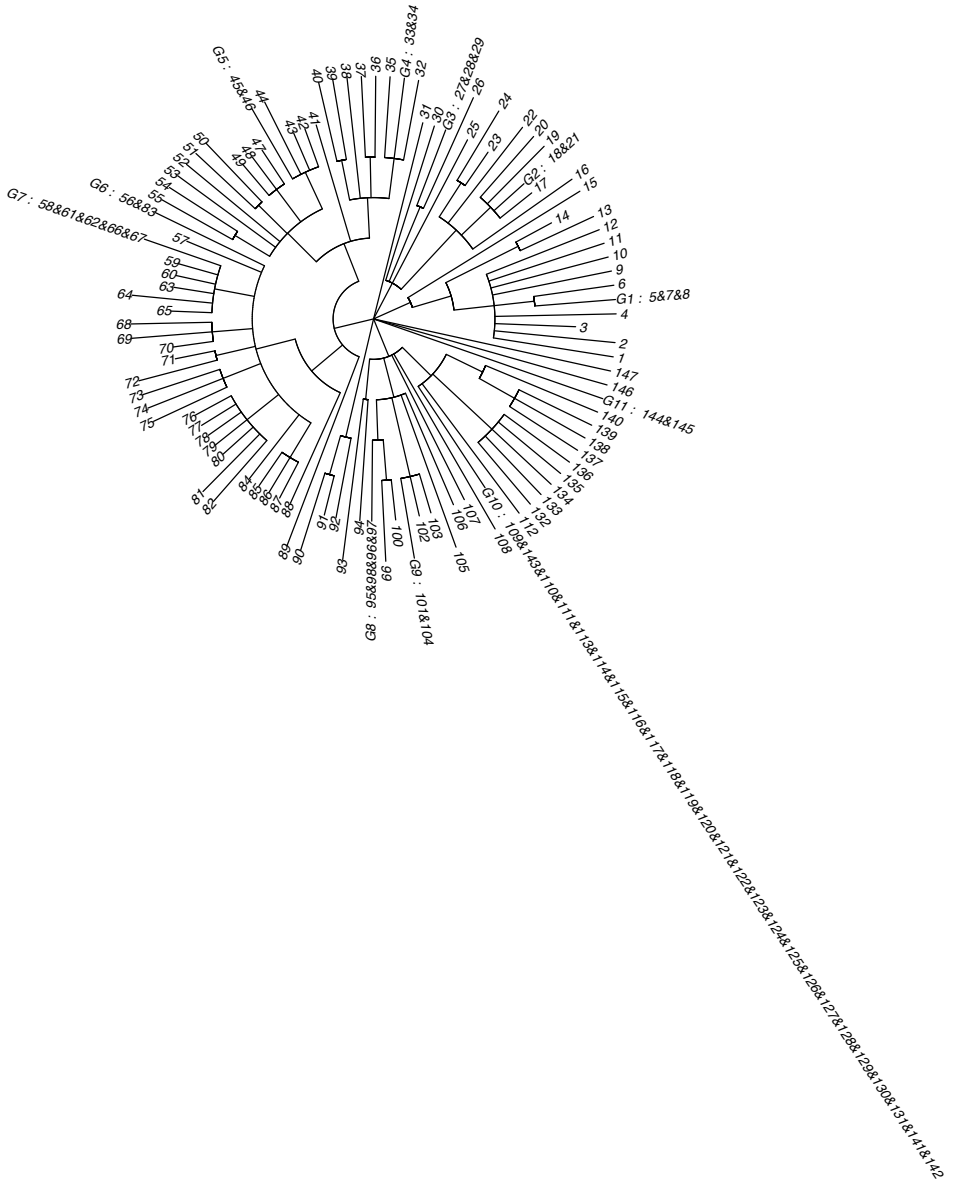


FIGURE 4.8: Phylogenetic tree for the Mother Buccal sample, after the application of the *thicking* and *liaising procedure*. Names and group IDs are listed in Annex Table 3.

CONCLUSIONS

In this work we have inspected some genomical and, more in general, biological applications with the advanced statistical techniques of functional data analysis, trying to combine the high dimensional setting of the genomic area with the complex structure of functional data. Specifically, we have provided new and advanced, but still computationally efficient, methods and we have converted them into user friendly and publicly available codes to allow other scientists to employ them, but also make fruitful improvements. Temporal clustering, FunChIP and FLAME have induced several biological insights, allowing to identify slower and faster decliners within Alzheimer's sufferers, or to relate ChIP-seq profiles to genomic locations and motif detection, or to detect a new SNP connected to lung underdevelopment of children or even to identify some Genus level abundances of the stool and buccal microbiome affecting the child's growth. However, many further developments can be carried on, both from the application point of view and from the statistical methodology aspect.

Focusing on further applications, we will investigate some developments for FunChIP, for Temporal clustering and for the microbiome project.
As for FunChIP, we aim to apply it not only to profiles of transcription factors, but also to new data like histone marks that are deposited on small regions of the genome, as promoters or enhancers (i.e. the histone marks H3K4me1, H3K4me3 or H3K27ac). Specifically, we want to inspect the presence of different shapes and their biological connections. Moreover, we aim to better investigate the connection between genomic locations and peak shapes, for example downsampling the MycER0h sample, to identify whether the shape of peaks remains invariant as well as the classification of genomic regions.
Focusing on Temporal Clustering, instead, we aim to generalize this method to other measures of cognitive ability, identifying the proper parametrization of these new longitudinal measurements.
As regards the microbiome application of the INSIGHT study, we are going to inspect, beside the microbiome effects, the influence of other covariates (related for example to "Child behavior", "Parenting", and "Family context") on the growth curves. These additional covariates can be treated independently from the the microbiome sets, or we can generalize FLAME to deal

with the mixed effects on the functional outcome.
Finally as regards the FLAME method, we aim to focus on many more applications, not only related to the biological area, but also to finance or geoscience, fileds characterized by high dimensionality, sparsity and complexity of data, as well as genomic. In fact, data collected from finance and geoscience are often very rough, but underlying parameters are believed to be very smooth, making FLAME an excellent candidate for the analysis.

Focusing on the methodological improvements, instead, we are planning to inspect more the LASSO penalty term of FLAME. For example, we can consider to mix the FLAME penalty with the classical Functional LASSO penalty, to identify whether the F-LASSO can control the shrinkage and a RKHS penalty can control the smoothing. Or, more in general, we can consider the possibility of introducing an elastic-net penalty to combine the LASSO and the ridge. Specifically, we aim to identify how these variations can affect both parameter selection and estimation.

To conclude, relevant insights on the genomic field have been uncovered. Nevertheless, the same methodologies could be applied in other areas of genomic, as well as in other fields, which show data with similar characteristics. At the same time, utilized methods could be further probed to deeply investigate their theoretical framework and their computational efficiency.

APPENDICES

A

FUNCHIP: A FUNCTIONAL DATA ANALYSIS APPROACH TO CLUSTER CHIP-SEQ PEAKS ACCORDING TO THEIR SHAPES

```
library(FunChIP)
```

A.1 INTRODUCTION

The `FunChIP` package provides a set of methods for the `GRanges` class of the package `GenomicRanges` to cluster ChIP-Seq peaks according to their shapes, starting from a `bam` file containing the aligned reads and a `GRanges` object with the corresponding enriched regions.

A.2 INPUT AND PREPROCESSING

ChIP-Seq enriched regions are provided by the user in a `GRanges` object `GR`. The user must provide the `bam` file containing the reads aligned on the positive and negative strands of the DNA. From the `bam` file we can compute, for each region of the `GRanges` (let N be the total number of regions), the base-level coverage separately for positive and negative reads. These two count vectors are used to estimate the distance d_{pn} between positive and negative reads and then the total length of the fragments of the ChIP-Seq experiment d. In particular, we assume that the positive and negative counts measure the same signal, shifted by d_{pn}, as they are computed from the two ends of the sequencing fragments. The global length of the fragment is the sum between the length of the reads of the `bam` file, r[1], and the distance between the positive and negative coverage d_{pn}

$$d = d_{pn} + r.$$

1 If in the `bam` file multiple length are present, r is estimated as the average length.

The function `compute_fragments_length` computes, from the `GRanges` object and the `bam` file, the estimated length of the fragments. Given a range for d_{pn}: $[d_{min}; d_{max}]$, the optimum distance d_{pn} is

$$d_{pn} = \text{argmin}_{\delta \in [d_{min}:d_{max}]} \sum_{n=1}^{N} D(f_{n+}, f^{\delta}_{n-}),$$

where f_{n+} is the positive coverage function of the n-th region, and f^{δ}_{n-} is the negative coverage of the n th region, shifted by δ. The distance D is the square of the L^2 distance between the coverages, normalized by the width of the region. The definition of the L^2 distance is detailed in Section A.4.

```
# load the GRanges object associated to the ChIP-Seq experiment
    on the transcription factor c-Myc in murine cells
data(GR100)

# name of the .bam file (the
# .bam.bai index file must also be present)
bamf <- system.file("extdata", "test.bam",
                    package="FunChIP", mustWork=TRUE)

# compute d
d <- compute_fragments_length(GR, bamf, min.d = 0, max.d = 300)
> estimated distance positive - negative read 148
> estimated read length 51
d
> 199
```

In Figure A.1 the distance function is shown varying the parameter δ, and the minimum value d_{pn} is computed.

Once we have correctly identified the fragment length we can compute the final coverage function to obtain the shape of the peaks. The `pileup_peak` method for the `GRanges` class uses the `bam` file to compute the base-level coverage on these regions, once the reads are extended up to their final length d. `pileup_peak` adds to the `GRanges` a `counts` metadata column, containing for each region a vector with length equal to the width of the region storing the coverage function.

```
# each peak of the GRanges object is associated to the
    correspondent coverage function

peaks <- pileup_peak(GR, bamf, d = d)
```

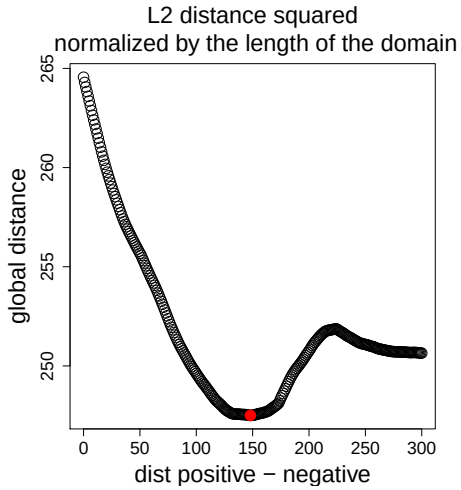


FIGURE A.1: Identification of d: optimal value of d_{pn} is shown. It is the minimum of the global distance function.

Additional information can be found in the help page of the `pileup_peak` method.

A.3 SMOOTHING

The `counts` metadata is approximated by a combination of splines to guarantee the smoothness and regularity needed for further analysis, as described in the following Sections.
The preprocessing steps carried out in the `smooth_peak` method are the following:

- *Removal of the background and extension.* In ChIP-Seq experiments, peaks may have an additive noisy background, and the removal of this background is mandatory to compare different peaks. The background is estimated as a constant value "raising" the peak and equal to the minimum value the coverage assumes. Consequently, once the background has been removed, each peak has zero as minimum value, thus allowing the peak to be indefinitely extended with zeros, if necessary. In Section A.4, how this choice affects the algorithm will be discussed.

- *Smoothing.* In order to be regular enough to computed derivatives, a peak has to be transformed in a suitable functional object, as described in Section A.4. The smoothing of the count vector c is performed through the projection of c on a cubic B-spline basis $\Phi = \{\phi_1, \dots \phi_K\}$

with a penalization on the second derivative as proposed by Ramsay and Silverman (2005). The result is a spline approximation of the data, which is continuous on the whole domain, together with its first order derivatives. Moreover, the penalization on the second derivative allows to control the global regularity of the function avoiding over-fitting and a consequent noisy spline definition. The spline approximation $s = \sum_{k=1}^{K} \theta_k \phi_k$ of the count vector $c = \{c_j\}$ is defined minimizing

$$S(\lambda) = \sum_{j=1}^{n} \left[c_j - s(x_j)\right]^2 + \lambda \int \left[s''(x)\right]^2 dx,$$

with x_j being the relative genomic coordinate the counts. The multiplying coefficient λ quantifies the penalization on the second derivative and is chosen through the Generalized Cross Validation criteria. For each peak i the GCV_i index is computed with a leave-one-out cross validation

$$GCV_i = \left(\frac{n}{n - df(\lambda)}\right)\left(\frac{SSE_i}{n - df(\lambda)}\right)$$

and then it is summed on the whole data set to obtain the global GCV. The number of degrees of freedom $df(\lambda)$ is automatically computed from the definition of the basis Φ.
The error SSE_i can be computed either on the data (SSE_i^0) or on the derivatives (SSE_i^1), to control the regularity of the function or the regularity of the derivatives, respectively:

$$SSE_i^0 = \sqrt{\sum_{j=1}^{n} \left(c_j - s(x_j)\right)^2} \text{ or } SSE_i^1 = \sqrt{\sum_{j=1}^{n-1} \left(\nabla c_j - s'(x_j)\right)^2},$$

with ∇c_j being the finite-difference approximation of the derivative of the `counts` vector c for the data i: $c = c(i)$, while $s'(x_i)$ is the evaluation of the first derivative $s' = s'(i)$ on the genomic coordinates. For further details on the spline definition see the `spline` function of the `fda` package Ramsay et al. (2014).

- *Scaling of the peaks.* This optional preprocessing step makes all the curves having the same width and area. In particular all the abscissa grid are scaled to become equal to the smallest grid throughout the data, while $y-$values are scaled to make areas of all the curves equal to 1.

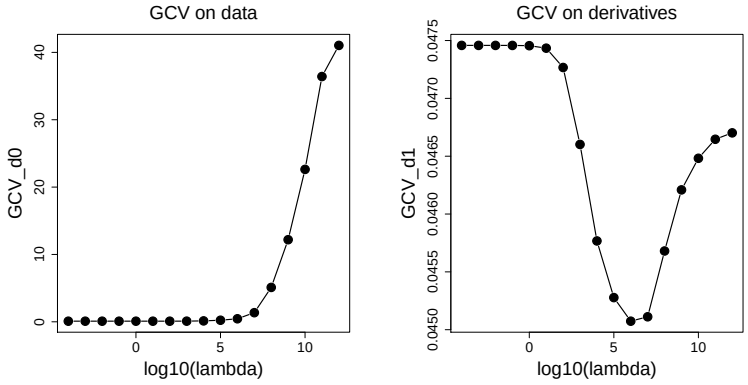


FIGURE A.2: The Generalized Cross Validation index computed on data (left), and on the derivatives (right), as a function of λ.

The `smooth_peak` method approximates the `counts` metadata by removing the background, computing the spline and potentially defining the scaled approximation. Focusing on the spline approximation, `smooth_peak` automatically chooses the optimal λ parameter according to the GCV criteria; the user can decide whether to consider the data or the derivatives to compute the SSE.

```
# the method smooth_peak removes the background and defines the
    spline approximation from the previously computed peaks with
    lambda estimated from the GCV on derivatives. The method
    spans a non-uniform grid for lambda from 10^-4 to 10^12. (
    the grid is uniform for log10(lambda) )
peaks.smooth <- smooth_peak(peaks, lambda = 10^(-4:12),
                        subsample.data = 50,
                        GCV.derivatives = TRUE,
                        plot.GCV = TRUE, rescale = FALSE )
```

In Figure A.2, the plot of the GCV for both data and derivatives is shown. From this Figure we see that the optimum value of λ, which minimizes the GCV for the derivatives, is also associated to a small value of the GCV for the data thus supporting the automatic choice.

```
# the automatic choice is lambda = 10^6
peaks.smooth <- smooth_peak(peaks, lambda = 10^6,
                        plot.GCV = FALSE)

# mantaining this choice of lambda smooth_peak can also define
    the scaled approximation of the spline
```

```
peaks.smooth.scaled <- smooth_peak(peaks, lambda = 10^6,
                                   plot.GCV = FALSE, rescale = TRUE)
```

Now the GRanges object contains, besides counts, 5 new metadata columns with the spline approximation evaluated on the base-level grid, its derivatives, the width of the spline and the new starting and ending points (see Figure A.8). For a more detailed description of the metadata columns, see the help page of the smooth_peak method.

With the introduction of the smoothing, counts at the edges of the peak are connected with regularity to 0, and therefore new values different from zeros may be introduced. In order to maintain regularity, the grid is extended up to the new boundaries.

Adding to smooth_peak the option rescale = TRUE the method, beside the 5 metadata columns previously introduced, returns 2 more metadata columns with the scaled approximation of the spline and its derivatives.

Once the spline approximation is defined, the summit of the smoothed peak (or even of the scaled peak), i.e. of its spline approximation, can be detected. The summit will be used to initialize the peak alignment procedure, described in Section A.4, and it can either be a user-defined parameter, stored in a vector of the same length of the GR, or automatically computed as the maximum height of the spline. The summit is stored in the new metadata column summit_spline. If the rescale option is set to TRUE the summit of the scaled approximation is also returned in the metadata colum summit_spline_rescaled.

```
# peaks.summit identifies the maximum point of the smoothed peaks
peaks.summit <- summit_peak(peaks.smooth)

# peaks.summit can identify also the maximum point of the scaled
    approximation
peaks.summit.scaled <- summit_peak(peaks.smooth.scaled,
                                   rescale = TRUE)
```

A.4 THE K-MEAN ALIGNMENT ALGORITHM AND THE CLUSTER_PEAK METHOD

The k-mean alignment algorithm is an efficient method to classify functional data allowing for general transformation of abscissae Sangalli et al. (2010); this general method is implemented in the package Parodi et al. (2015) and

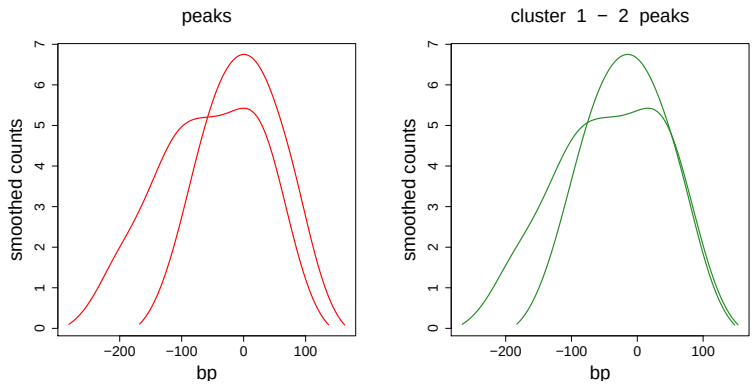


FIGURE A.3: Alignment procedure. Representation of two smoothed peaks. In the left panel they are not aligned, while in the right panel they are aligned with an integer shift.

various applications to real dataset are introduced in Sangalli et al. (2014), Bernardi et al. (2014), Patriarca et al. (2014).

In particular, given

- a set of curves $s_1, \ldots, s_n$,
- the number of clusters K,
- a distance function $d(s_i, s_j)$ between two curves s_i and s_j, as for example the integral of the difference $s_i - s_j$,
- a family of warping functions $\mathcal{W}$ to transform the abscissae of the curves and therefore align the peaks. Generally, $\mathcal{W}$ is the set of shifts or dilations or affine transformations (shift + dilation),

the algorithm, is an iterative procedure to split the curves into K clusters. The introduction of the warping function $h \in \mathcal{W}$ allows each curve to be shifted, dilated, or both, to define the minimum distance between curves. The new curve $s \circ h$ has the same values of s, but its abscissa grid is modified.

For example, in Figure A.3 two peaks are presented: in the left panel, they are not aligned, while the right panel shows the effects of alignment; the transformation of the abscissae (shift transformation) makes the two peaks more similar, and the distance d is not anymore affected by artificial phase distance. The code generating Figure A.3 calls `cluster_peak` and `plot_peak`, which are described in Section A.4.2 and Section A.5.

For the specific case of ChIP-Seq data, the admitted warping functions for the k-mean alignment algorithm (in the `cluster_peak` method), are integer shifts:

$$\mathcal{W} = \{h : h(t) = t + q \text{ with } q \in \mathbb{Z}\}. \tag{A.1}$$

In other words, with this choice, peaks can be shifted by integer values in the *alignment* procedure of the algorithm.

In the `cluster_peak` method the distance between two curves s_1 and s_2 is defined as

$$\begin{aligned} d(s_1, s_2) &= (1-\alpha)\, d_0(s_1, s_2) + \alpha w\, d_1(s_1, s_2) = \\ &= (1-\alpha)\, \|s_1^e - s_2^e\|_p + \alpha w\, \|(s_1^e)' - (s_2^e)'\|_p, \end{aligned} \tag{A.2}$$

where

- $\|f\|_p$ is the p norm of f. In particular, for $p = 0$, $\|\cdot\|_p$ is the L^∞ norm

  $$\|f\|_0 = \|f\|_{L^\infty} = \max_{x \in U} |f(x)|,$$

  with U being the domain of f.
  For $p = 1$, $\|\cdot\|_p$ is the L^1 norm

  $$\|f\|_1 = \|f\|_{L^1} = \int_U |f(x)| dx.$$

  And for $p = 2$, $\|\cdot\|_p$ is the L^2 norm

  $$\|f\|_2 = \|f\|_{L^2} = \int_U (f(x))^2\, dx.$$

- s_1^e and s_2^e are the functions s_1 and s_2 extended with zeros where not defined, after their backgrounds have been removed (see Section A.2). The distance function is computed on the union of the domains of s_1 and s_2 (U); s_1 and s_2 need to be extended to cover the whole U.

- $\alpha \in [0, 1]$ is a coefficient tuning the contributions of the norm of the data and the norm of the derivatives. If $\alpha = 0$, the distance is computed on the data, while if $\alpha = 1$ it is based on the derivatives. Intermediate values balance these two contributions: increasing the relevance given to the derivatives emphasizes the shapes of the peaks, while data are more related to the height.

- w is a weight coefficient, essential to make the norm of the data and of the derivatives comparable. It can be user defined or computed inside the `cluster_peak` method. A suggestion for computing the weight w is given in Section A.4.1.

A.4.1 *Definition of weight in the distance function*

If not provided, the method `cluster_peak` defines w as

$$w = \text{median}\left(\frac{d_0(s_i, s_j)}{d_1(s_i, s_j)}\right)$$

where $d_0(i,j) = \|s_i^e - s_j^e\|_p$ and $d_1(i,j) = \|(s_1^e)' - (s_2^e)'\|_p$. These matrices can be automatically computed with the `distance_peak` function.

```
# compute the weight from the first 10 peaks
dist_matrix <- distance_peak(peaks.summit)

# dist matrix contains the two matrices d_0(i,j)  and d_1(i,j),
    used to compute w
ratio_norm <- dist_matrix$dist_matrix_d0 /
                    dist_matrix$dist_matrix_d1
ratio_norm_upper_tri <- ratio_norm[upper.tri(ratio_norm)]
# suggestion: use the median as weight
w <- median(ratio_norm_upper_tri)
```

A.4.2 *The `cluster_peak` method*

The two main characteristics of the k-mean alignment algorithm used in `FunChIP` are the distance function `d` (defined in Equation (A.2)), used to compute the distance between curves, and the set of warping functions $\mathcal{W}$ (defined in Equation (A.1)) considered for the alignment. The `cluster_peak` method applies the k-mean alignment algorithm with these specifications to the set of peaks stored in the `GRanges` object. In particular, the parameters `weight`[2], `alpha` and `p` define the distance used in the algorithm, while `t.max` sets the maximum shift of each peak in each iteration (in this particular case, `q` of Equation (A.1) does not vary in the whole $\mathbb{Z}$ but `q` $\in$

2 `weight` can be also set to `NULL` and it will be automatically computed as specified in Section A.4.1. To save computational time, it is generally computed on a random sub-sample of data, whose size is set by the `subsample.weight` parameter.

$\{-$`t.max` $\cdot |U|, \ldots, +$`t.max` $\cdot |U|\}$, with $|U|$ being the maximum width of the spline approximation of the peaks.

Given a `GRanges` `GR` containing the metadata columns computed from the `smooth_peak` method, `cluster_peak` applies the k-mean alignment algorithm for all the values of `k` between `1` and `n.clust` (parameter of the function).

The algorithm can be run in parallel, setting to `TRUE` the `parallel` argument of the method and providing the number of cores `num.cores`. With these settings, the different applications of the algorithm, corresponding to different numbers of clusters, are executed in parallel.

As detailed in the help, the `cluster_peak` method has `2` outputs:

- The `GRanges` object, updated with new metadata columns associated to the classification. In particular, in the general case of classification with and without alignment, columns with information on the clustering of the peaks (`cluster_shift` and `cluster_NOshift`), the corresponding shifts (`coef_shift`) and the distances from the template of the clusters (`dist_shift` and `dist_NOshift`) are added.

- The graph of the global distance within clusters[3] as a function of the number of clusters (if `plot.graph.k = TRUE`). This plot can be used to identify the optimal number of clusters of the partition of the data set and the effect of the alignment procedure. In particular, if `shift = NULL`, the algorithm is run both with and without alignment and two trend lines are plotted: the black line corresponds to the global distance without the shift, and the red line corresponds to the distance obtained with alignment. If `shift` is set to `TRUE` or `FALSE`, just one type of algorithm is run and the correspondent curve is plotted. For each trend line, this graph allows the identification of the optimal value of the number of clusters: for this value, the distance significantly decreases with respect to the lower values of `k`, and negligibly increases with respect to higher values of `k` (elbow in the line). The gap between the red and the black line, instead, shows the decrease of the distance when the shift is introduced.

It is relevant to point out that the algorithm can be run both on the original data and on the scaled peaks, depending on the focus of the analysis. The logic paramter `rescale` allows the user to choose.

```
# classification of the smooth peaks in different numbers of
    clusters, from 1 ( no distinction, only shift ) to 6.
```

3 sum over all the peaks of the distance of each peak from the corresponding template.

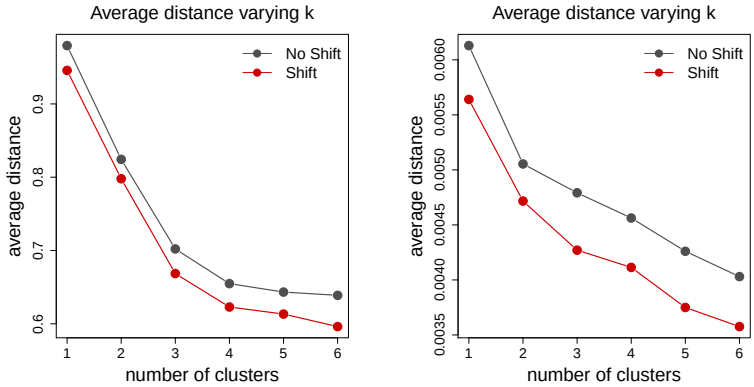


FIGURE A.4: **Global distance within clusters.** Global distance of the peaks form the corresponding template, as a function of the number of clusters k. In the left panel the graph for the original spline approximation, while in the right panel restuls are relative to the scaled approximation.

```
# here the analysis is run on the spline approximation without
    scaling
peaks.cluster <- cluster_peak(peaks.summit, parallel = FALSE,
                        seeds=1:6, n.clust = 1:6,
                        shift = NULL, weight = 1,
                        alpha = 1,  p = 2,
                        t.max = 2,  plot.graph.k = TRUE,
                        verbose = FALSE)

# here the analysis is run on the spline approximation with
    scaling
peaks.cluster.scaled <- cluster_peak(peaks.summit.scaled,
                              parallel = FALSE ,
                              seeds=1:6, n.clust = 1:6,
                              shift = NULL,  weight = 1,
                              alpha = 1, p = 2, t.max = 2,
                              plot.graph.k = TRUE,
                              verbose = FALSE, rescale = TRUE)
```

The particular case of k-mean alignment with $k = 1$ clusters can be used to highlight the effects of the alignment of the peaks: no grouping is performed, just the shifts are computed. Therefore, the decrease of the global distance is solely due to a change of the abscissae of the functions, as Figure A.3

shows. Moreover, focusing for exemple on the first panel of Figure A.4, we can deduce that, for this case

- the alignment can effectively decrease the distance, for exemple for $k = 6$, the gap between red and black line is significant;

- the alignment may change the optimal k: looking at the black line, one would have chosen $k = 4$, while the red line suggests $k = 3$ is the best choice. With the introduction of the shifts, data which are originally different becomes more similar and therefore one less cluster is needed; it has to be noted that the distance obtained with $k = 3$ and alignment is very similar to the one obtained with $k = 4$ and no alignment.

Therefore, for this case, one possible classification is the one associated to $k = 3$ with shift. On the contray for the scaled peaks the value of k we can identify as crucial is $k = 2$ and shift is relevant since it reduces a lot the global distance. The results for this specific number of clusters can then be selected with the `choose_k` method:

```
# select the results for k = 3 with alignment
peaks.classified.short <- choose_k(peaks.cluster, k = 3,
                            shift = TRUE, cleaning = TRUE)
peaks.classified.extended <- choose_k(peaks.cluster, k = 3,
                               shift = TRUE, cleaning = FALSE)

# and for the scaled version for k =2 and alignment
peaks.classified.scaled.short <- choose_k(peaks.cluster.scaled,
                                  k = 2, shift = TRUE,
                                  cleaning = TRUE)
peaks.classified.scaled.extended <-choose_k(peaks.cluster.scaled,
                                     k = 2, shift = TRUE,
                                     cleaning = FALSE)
```

The `choose_k` method allows, respectively, to remove all the metadata columns computed by `FunChIP` and obtain a `GRanges` equivalent to the initial one, with an extra the metadata column `cluster` containing the classification labels (`cleaning = TRUE`), or a `GRanges` retaining all the details of the pre-possessing and clustering (all the previously described metadata columns), with the extra column `cluster` (`cleaning = FALSE`).

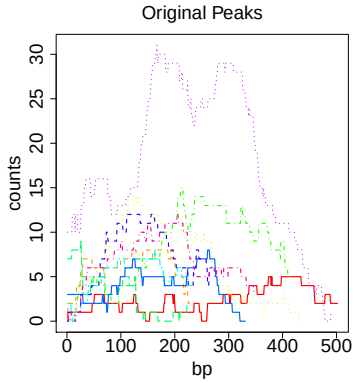


FIGURE A.5: Representation of 10 original peaks as raw counts (no smoothing)

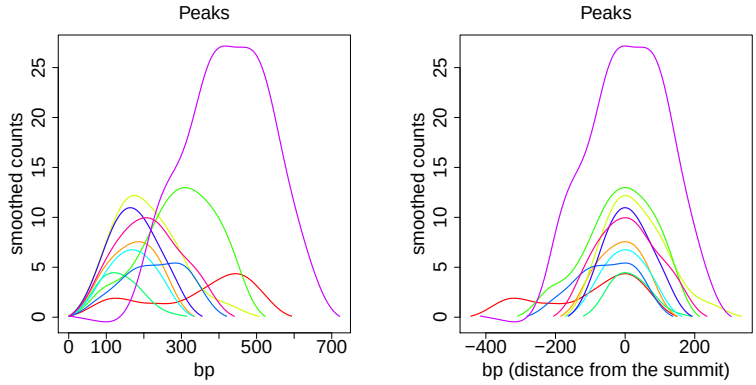


FIGURE A.6: In the left panel, the smoothed representation of the 10 peaks is shown, while in the right panel the same peaks are centered around their summits.

A.5 VISUALIZATION OF THE PEAKS

The `plot_peak` method is a very flexible function for displaying ChIP-Seq peaks. In particular, it allows to plot the raw counts obtained by the method `pileup_peak` , as in Figure A.5.

It can also plot smoothed peaks, possibly centered around the summit, as in Figure A.6, or scaled as in Figure A.7 and centerd.

From the comparison of Figure A.6 and Figure A.7 it is clear how the scaling affects the shape of splines. Now peaks are no more related to the magnitude, but just to their shapes.

Moreover, plotting both raw counts and spline is also possible: Figure A.8 shows a single peak in its raw and smoothed version. This representation is

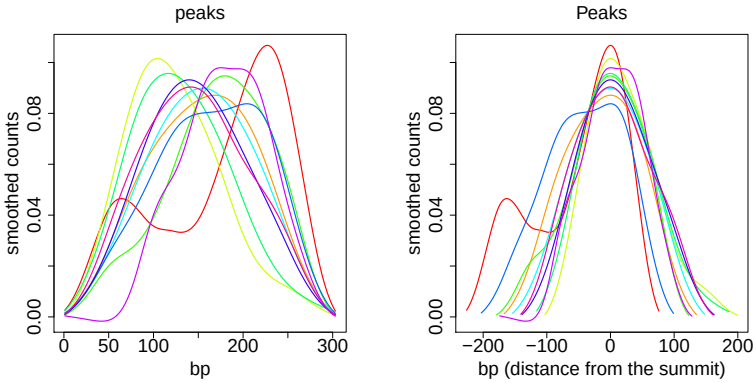


FIGURE A.7: In the left panel, the 10 scaled peaks are shown, while in the right panel the same peaks are centered around their summits.

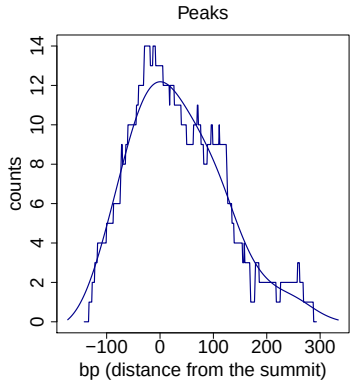


FIGURE A.8: Plot of the original read coverage of a peak and its smoothing (spline approximation), centered around the summit.

useful to check the accuracy of the smoothing and, if needed, manually set the λ parameter of the spline approximation.

Finally, the `plot_peak` method allows to plot the results of the clustering via the k-mean alignment. In Figure A.9 and Figure A.10, smoothed and scaled peaks are divided into the three clusters and plotted with the optimal shift obtained with the alignment.

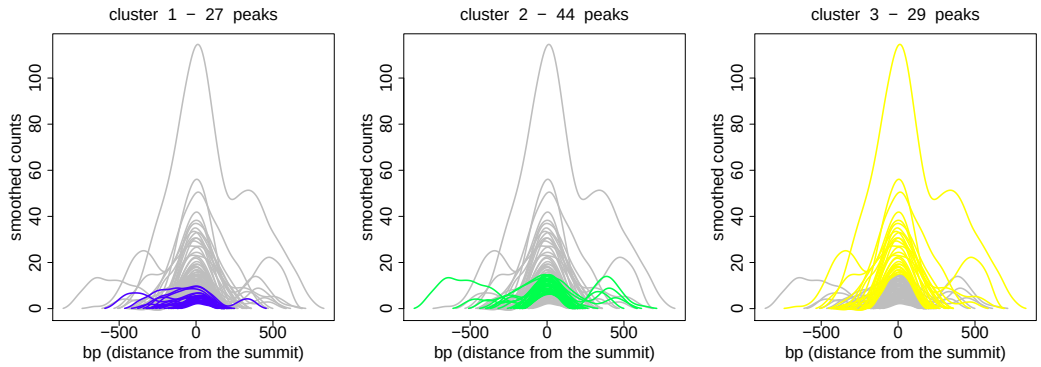


FIGURE A.9: Peaks divided in the three clusters: the same spline-smoothed peaks are plotted in grey, and for each panel the peaks in the corresponding cluster are colored to show their different shapes. Peaks are aligned with the shift coefficients obtained by the k-mean alignment algorithm.

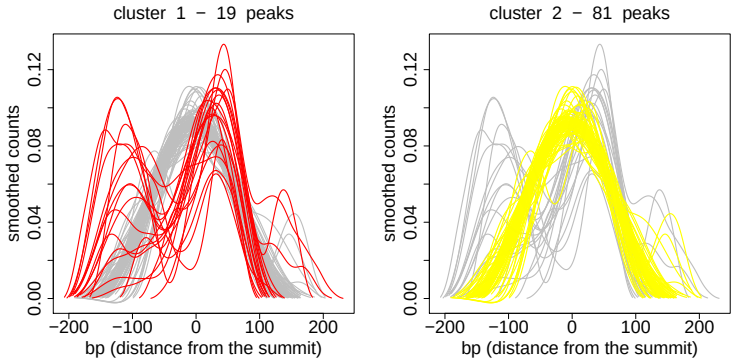


FIGURE A.10: Scaled peaks divided in the three clusters: The same spline-smoothed scaled peaks are plotted in grey, and for each panel the peaks in the corresponding cluster are colored to show their different shapes. Peaks are aligned with the shift coefficients obtained by the k-mean alignment algorithm.

B

FLM: FLAME ESTIMATION FOR HIGH DIMENSIONAL FUNCTION ON SCALAR REGRESSION

```
library(flm)
```

B.1 INTRODUCTION

The `flm` package provides an efficient tool to deal with Function-on-Scalar regression problems, mainly when the number of predictors `I` is much larger than the number of statistical units `N`. The main function of `flm` is `FLAME` that detects the set of significant predictrs and estimates their coefficients with the FLAME method. FLAME, *functional linear adaptive mixed estimation* is a methodology that simultaneously exploits the smoothness of the functional parameters as well as the sparsity of the predictors.

The Function-on-Scalar regression problem that FLAME aims to solve is

$$Y_n = \sum_{i=1}^{I} X_{n,i}\beta_i^\star + \varepsilon_n,$$

where $Y_1, \ldots, Y_N$ are independent random elements of a general Hilbert space $\mathbb{H}$, $\mathbf{X} = \{X_{n,i}\} \in \mathbb{R}^{N\times I}$ is a deterministic design matrix with standardized columns and ε_n are i.i.d. Gaussian random elements of $\mathbb{H}$ such that ε_n have 0 mean and covariance operator C.

Then, given the response functions and the predictors, the function `FLAME` can automatically identify the significant predictors and define the coefficients in a proper Reproducing Kernel Hilbert Space.

In Section B.2 and B.3 the detailed procedure of the estimation, from the definition of the kernel, using the functions `generation_kernel` and `generation_kernel_periodic`, to the solution of the Function-on-Scalar regression problem of the `estimation_beta` function, with some details on the

algorithm implementation. In Section B.4, instead, an example of an automatic usage of the package through the introduction of the `FLAME` function.

B.2 DEFINTION OF THE KERNEL

The main advantage of FLAME is the possibility of controlling the smoothness of the parameter estimates with the definition of a proper Reproducing Kernel Hilbert Space.

`flm` has two functions to define different RKHSs: `generation_kernel` and `generation_kernel_periodic` define eigenvalues θ_j, eigenfunctions v_j and their derivatives of a kernel K. Then K is, for the spectral theorem of Dunford and Schwartz (1963)

$$K = \sum_{j=1}^{\infty} \theta_j v_j \otimes v_j$$

The kernel we examine in this package are the Sobolev, the Exponential, the Gaussian (Section B.2.1) and the Periodic kernel (Section B.2.2).

B.2.1 *Exponential, Sobolev and Gaussian kernel*

The `generation_kernel` function allows the user to define the Exponential, the Sobolev and the Gaussian kernel.

Here an explicit definition of the three kernels:

- **Sobelev kernel**: Consider $\mathbb{H} = L^2(\mathcal{D})$, where $\mathcal{D}$ is a compact subset of $\mathbb{R}^d$. We can define $\mathbb{K}$ to be the subset of functions in $L^2(\mathcal{D})$ that have up to and including m^{th} order derivatives that are also in $L^2(\mathcal{D})$. In this package we limit our analysis to $m = 1$ and $d = 1$. Then, we define a family of norms on $\mathbb{K}$ as

$$\|x\|_{\mathbb{K}}^2 = \int_{\mathcal{D}} |x(s)|^2 \, ds + \frac{1}{\sigma} \int_{\mathcal{D}} |x'(s)|^2 \, ds;$$

  here the σ parameter controls the influence of the H^1 norm and then the smoothness of the eigenfunctions. Increasing σ the smoothness level decreases. Equipped with this norm, $\mathbb{K}$ is an RKHS if and only if $m > d/2$, as in our case of one-dimensional functions ($d = 1$) in H^1

($m = 1$). The kernel cannot always be written down explicitly, but in the case where $\mathcal{D} = [0, 1]$ and $m = 1$, we have that

$$K(t, s) = \begin{cases} \frac{\sigma}{\sinh(\sigma)} \cosh(\sigma(1-s)) \cosh(\sigma t) & t \leqslant s \\ \frac{\sigma}{\sinh(\sigma)} \cosh(\sigma(1-t)) \cosh(\sigma s) & t > s \end{cases} .$$

Then we can numerically solve the equation to isolate the eigenfunctions and the eigenvalues of K as the `sobolev_kernel` function does. This function is implicitly called in `generation_kernel`. Details on the Sobolev kernel can be found in Berlinet and Thomas-Agnan (2011).

- **Gaussian Kernel** Let $\mathbb{H} = L^2(\mathcal{D})$, with $\mathcal{D}$ a compact subset of $\mathbb{R}^d$ $\forall d$. The Gaussian kernel if $d = 1$ is given by

$$K(s, s') = \exp\left\{-\sigma |s - s'|^2\right\}.$$

While the Sobolev spaces contain functions which are differentiable up to a given order, the space $\mathbb{K}$ here contains functions which are infinitely differentiable. When used in FLAME, such a kernel produces very smooth estimates. As for the Sobolev kernel, the smoothness level of the kernel is controlled by the σ parameter. Increasing σ the smoothness level is reduced and FLAME get a more rough estimates. The definition of the kernel function is coded in the `kernlab` R package of Karatzoglou et al. (2004).

- **Exponential Kernel**: The exponential kernel is on the other end of the "smoothness" spectrum compared to the Gaussian kernel. In the one-dimensional case we have

$$K(s, s') = \exp\left\{-\sigma |s - s'|\right\}.$$

This seemingly minor adjustment to the power in the exponent produces a space consisting of continuous functions which need not to be differentiable. Using this kernel produces substantially rougher FLAME estimates than the Gaussian kernel. They are also a bit rougher than the Sobolev kernel as well. As for the previous kernels, the smoothness parameter σ tunes the regularity level of the FLAME estimations. And as for the Gaussian kernel the `kernlab` R package of Karatzoglou et al. (2004) provides an explicit definition of the kernel matrix.

The `generation_kernel` function, then, allows the user to define the eigenfunctions and eigenvalues of these three different kernels, once the time domain is defined in the `domain` argument. The `type_kernel` parameter defines the type of kernel: `'exponential'`, `'sobolev'` and `'gaussian'` are the three possible choices; the `param_kernel` argument, instead, is the σ parameter tuning the regularity level. The number of eigenfunctions ν_j (which define the basis functions of the RKHS) is chosen as

$$\sum_{j=1}^{J} \theta_j \geqslant \text{thres} \sum_{j=1}^{\infty} \theta_j.$$

where the `thres` parameter is an input of the `generation_kernel` function and θ_j are the eigenvalues of the kernel.

In the following chunk an example of definition of Sobolev kernel with $\sigma = 8$ and in Figure B.1 the first four eigenfunctions and their derivatives, with the correspondent ratio of explained variability $\theta_j / \sum_j \theta_j$.

```
type_kernel <- "sobolev"
param_kernel <- 8
M <- 50
T_domain <- seq(0, 1, length = M)  # time point grid.
thres <- 0.99 # thresold for the eigenvalues.
kernel_here <- generation_kernel(type = type_kernel,
                          param = param_kernel,
                          domain = T_domain,
                          thres = 0.99,
                          return.derivatives = TRUE)
eigenval <- kernel_here$eigenval
eigenvect <- kernel_here$eigenvect
derivatives <- kernel_here$derivatives
```

B.2.2 *Periodic kernel*

A very useful feature of working with an RKHS is that one can also include periodicity and boundary conditions into the parameter estimates, using the `generation_kernel_periodic` function, for example, the user can define a kernel with a fixed periodicity p and a smoothing parameter σ. If you have yearly measurements with seasonal or semestral periodicity, for example, you may use the periodic kernel with period $p = 1/4$ or $p = 1/2$.

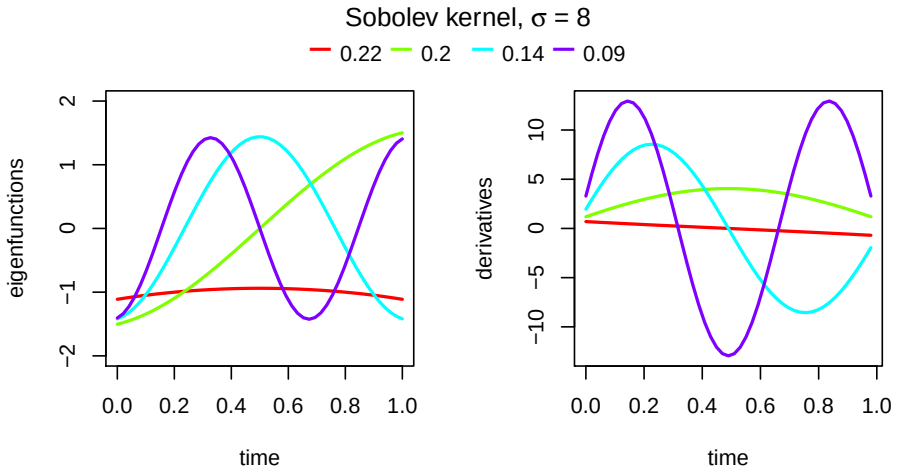


FIGURE B.1: Plot of the first 4 eigenfunctions (left panel) and derivatives (right panel) of the Sobolev kernel with parameter $\sigma = 8$. The correspondent explained variance is on the top of the plot.

The kernel for period p on a one dimensional domain is defined as

$$K(s, s') = \sigma^2 \exp\left\{-2/\sigma \sin^2\left(\frac{\pi|s - s'|}{p}\right)\right\}.$$

In the following chunk an example of definition of periodic kernel with `period = 1/2` and in Figure B.2 the first four eigenfunctions and their derivatives, with the correspondent ratio of explained variability.

```
param_kernel <- 8
M <- 50
T_domain <- seq(0, 1, length = M)
kernel_here <- generation_kernel_periodic(period = 1/2,
                                          param = param_kernel,
                                          domain = T_domain,
                                          thres = 1-10^{-16},
                                          return.derivatives = TRUE)
eigenval <- kernel_here$eigenval
eigenvect <- kernel_here$eigenvect
derivatives <- kernel_here$derivatives
```

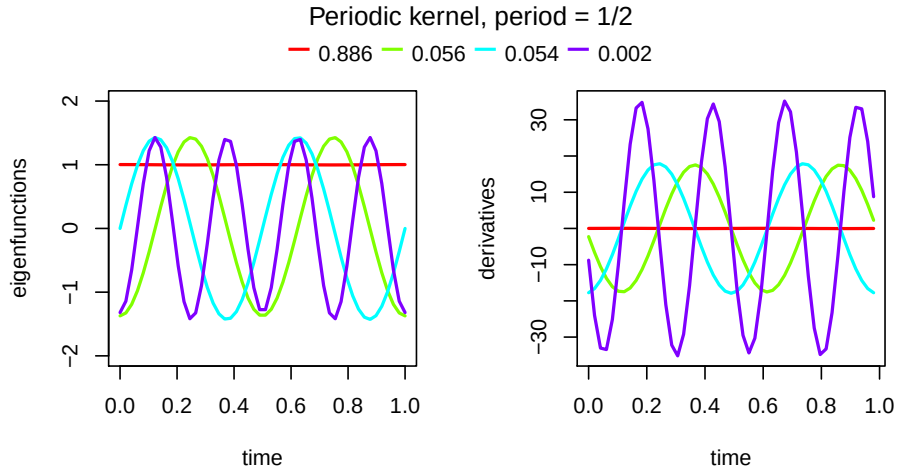


FIGURE B.2: Plot of the first 4 eigenfunctions (left panel) and erivatives (right panel) of the periodic kernel with period $p = 1/2$. The correspondent explained variance is on the top of the plot.

B.3 FLAME ESTIMATION

In this section we define an example of generation of data for a Function-on-Scalar linear model (Section B.3.1), we present the kernel for the estimation (Section B.3.2) and an outline of the FLAME method (Section B.3.3) with an analysis of the results (Section B.3.4).

B.3.1 *Generation of data*

We define an high-dimensional setting simulation with `N = 500` and `I = 1000` to highlight both the efficiency of FLAME in the estimation and in variable selection. Only `I0 = 10` predictors, in fact, are meaningful for the response, the others have null effect on the Y's.

The predictor matrix $\mathbf{X}$ is the standardized version of a matrix randomly sampled from a N dimension Gaussian distribution with 0 average and covariance C. The true coefficients $\beta^\star(t)$ are sampled from a Matèrn process with 0 average and parameters $(\nu = 2.5, \text{range} = 1/4, \sigma^2 = 1)$.

Observations $y(t)$ are, then, obtained as the sum of the contribution of all the predictors and a random noise, a 0-mean Matèrn process with parameters $(\nu = 1.5, \text{range} = 1/4, \sigma^2 = 1)$. Functions are sampled on a $m = 50$ points grid.

The Matèrn covariance operator is defined in the `covMaterniso` function.

In Figure B.3 the plot of the coefficients $\beta^*(t)$, 20 random errors $\varepsilon(t)$ and the correspondent response functions $Y(t)$.

```
N <- 500 # number of data
I <- 1000 # number of predictors
I0 <- 10 # number of non-zero predictors

set.seed(16589)

# definition of the time domain
m <- 50 # total number of points
T_domain <- seq(0, 1, length = m) # time points, length = m
M_integ <- length(T_domain)/diff(range(T_domain)) # coefficient
    for the computation of the integrals

# definition of the design matrix X, in this specific case the
    covariance matrix C is the identity matrix
mu_x <- rep(0, I)
C <- diag(I)
X <- mvrnorm(n=N, mu=mu_x, Sigma=C)
X <- scale(X) # normalization

# definition of the coefficients
nu_beta <- 2.5
range <-1/4
variance <- 1
hyp <- c(log(range), log(variance)/2) # set of parameters for the
     Matrn Covariance operator of beta
mu_beta <- rep(0,m) # mean of the beta
Sig_beta <- covMaterniso(2*nu_beta, hyp, T_domain)
beta <- mvrnorm(mu=mu_beta, Sigma=Sig_beta, n=I0) # generation of
     the I0 significant coefficients

# definition of the randomerrors
nu_eps <- 1.5
mu_eps <- rep(0, m)
Sig_eps <- covMaterniso(2*nu_eps, hyp, T_domain)
eps <- mvrnorm(mu=mu_eps, Sigma=Sig_eps, n=N) # generation of the
     N random errors

I_X <- sort(sample(1:I, I0)) # index of the I0 significant
    predictors
```

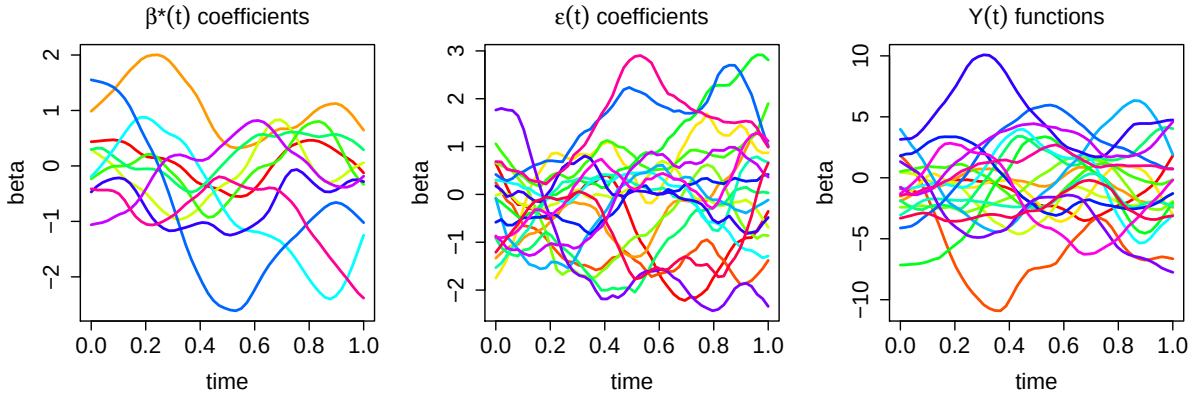


FIGURE B.3: Random generation of data. From the left, 10 coefficients $\beta^*(t)$, 20 random errors $\varepsilon(t)$ and the correspondent 20 response functions $Y_n(t)$

```
Y_true <- X[,I_X] %*% beta
Y_full <- X[,I_X] %*% beta + eps # Y_n observations
```

B.3.2 *Defintion of the kernel and projection*

For this simulation we choose as kernel the Sobolev kernel with $\sigma = 8$ and a threshold for the eigenvalues 0.99. The eigenfunctions of the kernel are an orthogonal basis both for the space $\mathbb{H}$ and for $\mathbb{K}$; then for the following estimation we can project the $Y_n(t)$ functions on that basis with the `projection_basis` function.

```
# defintion of the kernel
type_kernel <- "sobolev"
param_kernel <- 8
m <- 50
T_domain <- seq(0, 1, length = m) # time domain
thres <- 0.99
kernel_here <- generation_kernel(type = type_kernel,
                                 param = param_kernel,
                                 domain = T_domain,
                                 thres = 0.99,
                                 return.derivatives = TRUE)
eigenval <- kernel_here$eigenval
eigenvect <- kernel_here$eigenvect
derivatives <- kernel_here$derivatives

# preojection on the kernel basis of y and beta
```

```
Y_matrix <- projection_basis(Y_full, eigenvect, M_integ)
B_true <- projection_basis(beta, eigenvect, M_integ)

matrix_beta_true_full <- matrix(0, dim(B_true)[1], I)
matrix_beta_true_full[, I_X] <- B_true
```

Given the definition of the derivatives of the eigenfunctions of the kernel (returned by the `generation_kernel` function), we can also define the derivatives of the true coefficients $\beta^\star$ and of the responses.

```
B_true_der <- t(kernel_here$derivatives %*% B_true)

Y_true_der <- X[,I_X] %*% B_true_der
```

B.3.3 *The FLAME method*

The function `estimation_beta` allows the user to compute the FLAME estimation. The back-end of this function is written in `c++` (and available in the `FLAME_functions_cpp.cpp` function), so that the computation is efficient also in the high dimensional setting.

The function mainly consist of a coordinate-descent algorithm to define the FLAME estimation minimizing the target function

$$\begin{aligned} L(\beta) &= \frac{1}{2N}\sum_{n=1}^{N}\|Y_n - X_n\beta\|_{\mathbb{H}}^2 + \lambda\sum_{i=1}^{I}\tilde{\omega}_i\|\beta_i\|_{\mathbb{K}} = \\ &= \frac{1}{2N}\|Y - \mathbf{X}\beta\|_{\mathbb{H}}^2 + \lambda\sum_{i=1}^{I}\tilde{\omega}_i\|\beta_i\|_{\mathbb{K}} \end{aligned}$$

with $Y \in \mathbb{H}^N$, $\mathbf{X} \in \mathbb{R}^{N\times I}$ and $X_n = \mathbf{X}_{(n,\cdot)} \in \mathbb{R}^I$, $\beta \in \mathbb{K}^I$. Throughout, we use notation such as $\mathbb{H}^N$ to denote product spaces. For the sake of simplicity, we abuse notation by letting $\|\cdot\|_{\mathbb{H}}$ also denote the induced Hilbert space norm on product spaces such as $\mathbb{H}^N$.

The $\tilde{\omega}_i$ parameters are used to balance the contribution of the different coefficients and to make the LASSO estimator unbiased. The function `estimation_beta` has the estimation of $\tilde{\omega}_i$ as first objective. The coordinate-descent algorithm, in fact is run twice. The first one, the *non adaptive step*, is

run defining as 1 all the weights $\tilde{\omega}_i$ and the second one, the *adaptive step*, is run, to obtain an unbiased estimator, with

$$\tilde{\omega}_i = \frac{1}{\|\hat{\beta}_i^1\|_{\mathbb{K}}},$$

where $\hat{\beta}_i^1$ are the estimated coefficient of the *non-adaptive step*.

A key parameter for the estimation is λ, used to balance the prediction error $\|Y - \mathbf{X}\beta\|_{\mathbb{H}}^2$ and the smoothness level of the estimations $\sum_{i=1}^{I} \tilde{\omega}_i \|\beta_i\|_{\mathbb{K}}$. The two steps of the algorithm are both run on a grid of λ and the best value is chosen with a cross-validation criteria, selecting a training set, made up by the `proportion_test_set` percent of the data, and the remaining test set. The `estimation_beta` function automatically defines the grid for the λ parameter in the two runs as a logarithmic evenly gird from a maximum value, λ_{max}

$$\lambda_{max} = \max_{i=1,\dots,I} \omega_i^{-1} \|N^{-1} \sum X_n^i K(Y\_n)\|_{\mathbb{K}}$$

to the minimum value `ratio_lambda` $\cdot \lambda_{max}$. The user, beside the `ratio_lambda` parameter can also define the length of the grid, in the `number_lambda` parameter.

Focusing on the coordinate-descent method. It is based on the subgradient equation

$$\begin{aligned}
\frac{\partial}{\partial \beta_i} L(\beta) &= -\frac{1}{N} \sum_{n=1}^{N} X_{n,i} K(Y_n - X_n^\top \beta) + \lambda \tilde{\omega}_i \begin{cases} \|\beta_i\|_{\mathbb{K}}^{-1} \beta_i, & \beta_i \neq 0 \\ \{h \in \mathbb{K} : \|h\|_{\mathbb{K}} \leqslant 1\}, & \beta_i = 0 \end{cases}. \\
&= -K(\tilde{\beta}) + K(\beta_i) + \lambda \omega_i \begin{cases} \|\beta_i\|_{\mathbb{K}}^{-1} \beta_i, & \beta_i \neq 0 \\ \{h \in \mathbb{K} : \|h\|_{\mathbb{K}} \leqslant 1\}, & \beta_i = 0 \end{cases}.
\end{aligned}$$

with $\tilde{\beta}$ the least squares estimator $\tilde{\beta}_i = \frac{1}{N} \sum_{n=1}^{N} X_{n,i} E_n$ where E_n is the residual $E_n = Y_n - \sum_{j \neq i} X_{n,j} \hat{\beta}_j$ and it is updated at each iteration. From the subgradient equation we can also detect the meaning of the maximum value for λ: λ_{max}, in fact, is the minimum value of λ for which all the predictors are guaranteed to have 0 coefficient. For all i

$$\|K(\tilde{\beta}_i\|_{\mathbb{K}} \leqslant \lambda \omega_i$$

```
FLAME <- estimation_beta(X = X,
```

```
                    Y = Y_matrix,
                    eigenval = eigenval,
                    NoI = 10,
                    thres = 0.1,
                    number_non_zeros = I0*2,
                    ratio_lambda = 0.01,
                    number_lambda = 100,
                    proportion_training_set = 0.75,
                    verbose = FALSE)
```

Moreover we present two features we have included in the algorithm to increase the computational efficiency. The first is a *warm start* which means that when moving to the next λ in the grid, we use the previous $\hat{\beta}$ as initial estimation and, due to the small changes in λ, this means that the new $\hat{\beta}$ can be computed very quickly. The second feature is the *kill switch* parameter. This allows the user to set the maximum number of significant predictors to be selected by the model: when the algorithm moves past this threshold, the algorithm is stopped.

The `estimation_beta` function automatically performs all this steps and returns both the final result after the *adaptive step* and the intermediate result, just after the *non adaptive step*.

To directly access to the coordinate descent method and perform manually the estimation, for example fixing a specific value for λ, a specific set of weights or a specific starting point for the estimated β, the user can run the `defintion_beta` function, the one that is implicitly called in `estimation_beta`.

We can notice that the `estimation_beta` function returns as the estimation of the coefficients $\hat{\beta}$ the matrix of their projection on the kernel basis. The function `projection_domain` allows to compute the estimation on the time domain and then to represent the results, as in Figure B.4. Here we show a comparison with the true simulated $\beta^\star$ functions.

```
beta_on_time_grid <- projection_domain(FLAME$beta, eigenvect)
y_on_grid_estimated <- X %*% beta_on_time_grid
```

B.3.4 *Analysis of the results*

To analyze the result of this simulation, first of all we detect the relevant predictors isolated by FLAME and we compare them with the true ones. We notice that FLAME correctly isolates the `I0 = 10` relevant predictors, without any false positive predictor added.

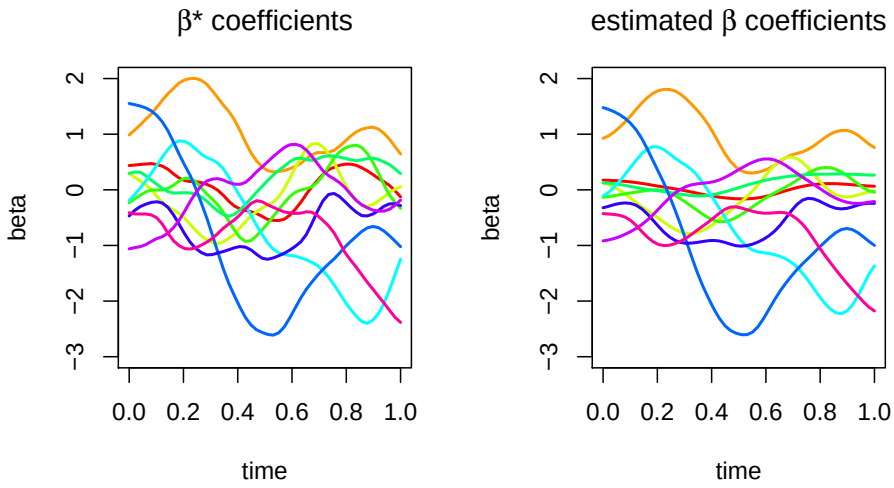


FIGURE B.4: In the left panel the plot of the simulated $\beta^{\star}$ coefficients, while in the right panel the FLAME coefficients $\hat{\beta}$ are shown.

```
I_X  # list of the true non zero predictors
> 18 97 287 297 433 527 642 709 901 934
FLAME$predictors # list of the estimated non zero predictors
> 18 97 287 297 433 527 642 709 901 934

true_positives <- length(which(I_X %in% FLAME$predictors))
true_positives # number of significant predictors correctly
    identified
> 10
false_positives <- length(which(!(FLAME$predictors %in% I_X)))
false_positives # number of non significant predictors wrongly
    picked by the algorithm
> 0
```

Then we introduce a short analysis of the result computing:

- the prediction error on data

$$\sum_{n=1}^{N} \| \mathbf{X}_n \beta^* - \mathbf{X}_n \hat{\beta} \|_{L^2},$$

- the prediction error on derivatives

$$\sum_{n=1}^{N} \| \mathbf{X}_n \beta^{*\prime} - \mathbf{X}_n \hat{\beta}' \|_{L^2},$$

- the $\mathbb{K}$-norm of the error in the prediction of the β coefficients

$$\sum_{i=1}^{I} \|\beta_i^* - \hat{\beta}_i\|_{\mathbb{K}}.$$

This last error can be easily computed with the `norm_matrix_K` function.

```
beta_der_on_grid_estimated<- kernel_here$derivatives \%*\% FLAME$
    beta
prediction_error - sum(apply(Y_true - y_on_grid_estimated, 1,
    function(x) {
        sqrt((2 * sum(x^2)-x[1]^2-x[length(x)]^2)/(M_integ *2)) }
      ))
prediction_error
> 218.4826

estimated_y_der_grid <- X %*% t(beta_der_on_grid_estimated)
prediction_error_der <- sum(apply(Y_true_der - estimated_y_der_
    grid, 1,  function(x) {
        sqrt((2 * sum(x^2)-x[1]^2-x[length(x)]^2)/(M_integ *2)) }
      ))
prediction_error_der
> 2687.767

norm_K_beta <- sum(norm_matrix_K(matrix_beta_true_full - FLAME$
    beta, eigenval)^2)
norm_K_beta
> 0.76914
```

B.4 THE AUTOMATIC USAGE OF FLAME FOR FUNCTION-ON-SCALAR REGRESSION PROBLEMS

In this final Section we present the `FLAME` function that allows the user a direct solution of the regression problem. From an `fd` object, or a point-wise evaluation of the response functions and the set of predictors, the function automatically detects the significant predictors and computes the estimation. It is possible to provide the kernel, choosing among `"exponential"`, `"gaussian"`, `"sobolev"` and `"periodic"` and fixing the smoothness parameter. Here an example of estimation with the predictors provided as an `fd`

object. The $y(t)$ of Section B.3.1 are represented as their projection on a 20 elements cubic Bspline basis and then also the estimated coefficients are returned as an `fd` object.

```
class(Y_fd)
estimation_auto <-  FLAME(Y_fd, # fd object for the response
                          X, # predictors matrix
                          number_non_zeros = 20)
# default choice for the kernel is Sobolev with sigma = 8,
names(estimation_auto)
>  "beta"       "predictors"
class(estimation_auto$beta)
>  "fd"
estimation_auto$predictors
> 18  97 287 297 433 527 642 709 901 934
```

ANNEX

LIST OF GROUPED ABUNDANCES FOR THE THREE MICROBIOME SAMPLES OF CHAPTER 4

ID	Child Buccal Microbiome: Genus level names
1	Actinomyces + Actinobaculum + Arcanobacterium + Candidatus_Ancillula + Mobiluncus + N09 + Trueperella + Varibaculum + Actinopolyspora + Saccharothrix + Lentzea + Kibdelosporangium + Actinokineospora + Actinoalloteichus + Georgenia + Brevibacterium + Demequina + Cellulomonas + Actinotalea
2	Corynebacterium + Cryptosporangium + Dermabacter + Brachybacterium + Dermacoccus + Dermatophilus + Dietzia + Frankia + Modestobacter + Geodermatophilus + Blastococcus + Glycomyces
3	Gordonia + Tetrasphaera + Terracoccus + Serinicoccus + Phycicoccus + Kytococcus + Knoellia + Janibacter + Arsenicicoccus + Jonesia + Kineosporia + Kineococcus
4	Microbacterium + Leucobacter + Frigoribacterium + Curtobacterium + Cryocola + Cryobacterium + Clavibacter + Candidatus_Rhodoluna + Candidatus_Aquiluna + Agromyces + Agrococcus + Mycetocola + Pseudoclavibacter + Rathayibacter + Salinibacterium
5	Kocuria + Citricoccus + Arthrobacter + Microbispora
6	Micrococcus + Nesterenkonia + Renibacterium
7	Rothia + Sinomonas + Zhihengliuella
8	Mycobacterium + Virgisporangium + Verrucosispora + Solwaraspora + Pilimelia + Dactylosporangium + Couchioplanes + Catellatospora + Actinoplanes + Actinocatenispora + Rhodococcus + Nocardia + Propionicimonas + Pimelobacter + Nocardioides + Kribbella + Friedmanniella + Aeromicrobium + Actinopolymorpha + Streptomonospora + Prauseria + Nocardiopsis + Xylanimicrobium + Promicromonospora + Cellulosimicrobium
9	Propionibacterium + Microlunatus + Luteococcus + Tessaracoccus + Thermobispora + Saccharopolyspora + Saccharomonospora + Pseudonocardia + Prauserella + Jiangella + Amycolatopsis + Actinomycetospora + Rarobacter + Sanguibacter + Sporichthya
10	Streptomyces + Kitasatospora + Streptosporangium + Nonomuraea + Actinomadura + Actinocorallia + Actinoallomurus + Tsukamurella + Williamsia + Yaniella
11	Bifidobacterium + Alloscardovia + Bombiscardovia + Gardnerella + Scardovia
12	Atopobium + Adlercreutzia + Collinsella + Coriobacterium + Eggerthella + Slackia + Candidatus_Microthrix + Iamia + Ferrimicrobium + Nitriliruptor + Euzebya + Rubrobacter + Solirubrobacter + Patulibacter + Conexibacter
13	Bacteroides + BF311 + X5_7N15
14	Candidatus_Azobacteroides
15	Dysgonomonas
16	Paludibacter
17	Parabacteroides
18	Porphyromonas
19	Tannerella
20	Prevotella + Rikenella + PW3 + Blvii28 + AF12
21	Odoribacter + Butyricimonas + BE24
22	Paraprevotella + CF231 + YRC22
23	[Prevotella]
24	Roseivirga + Reichenbachiella + Persicobacter + JTB248 + Fulvivirga + Flexithrix + Flexibacter + Flammeovirga + Sporocytophaga + Spirosoma + Runella + Rudanella + Rhodocytophaga + Pontibacter + Microscilla + Leadbetterella + Hymenobacter + Flectobacillus + Emticicia + Dyadobacter + Cytophaga + Adhaeribacter + Ucs1325 + SGUS912 + SC3-56 + Candidatus_Cardinium + Candidatus_Amoebophilus
25	Capnocytophaga + Arenibacter + Aquimarina + Aequorivita + Cellulophaga + Wandonia + Owenweeksia + Fluviicola + Cryomorpha + Crocinitomix + Brumimicrobium + Blattabacterium
26	Flavobacterium + Gelidibacter + Gillisia + Gramella + Kordia + Lacinutrix + Leeuwenhoekiella + Lutimonas + Maribacter + Mesonia + Muricauda + Myroides + Polaribacter + Psychroflexus + Psychroserpens + Robiginitalea + Salegentibacter + Salinimicrobium + Sediminicola + Tenacibaculum + Ulvibacter + Winogradskyella + Zhouia + Zobellia
27	Chryseobacterium
28 - G1	Cloacibacterium + Elizabethkingia + Ornithobacterium + Riemerella
29 - G1	Wautersiella + Weeksella
30	KSA1 + Balneola + Salisaeta + Salinibacter + Rubricoccus + Rhodothermus + Sphingobacterium + Pedobacter + Olivibacter + Saprospira + Lewinella + Haliscomenobacter + Segetibacter + Sediminibacterium + Niabella + Flavisolibacter + Flavihumibacter + Chitinophaga

ID	Child Buccal Microbiome: Genus level names
31	Synechococcus + Prochlorococcus + Thermosynechococcus + Acaryochloris + Pseudanabaena + Prochlorothrix + Leptolyngbya + Halomicronema + Arthronema + Symploca + Planktothrix + Planktothricoides + Phormidium + Oscillatoria + Microcoleus + Geitlerinema + Chroococcidiopsis + Spirulina + Prochloron + Microcystis + Cyanothece + Chroococcus + Rivularia + Calothrix + Nostoc + Gloeotrichia + Dolichospermum + Cylindrospermopsis + Anabaenopsis + Anabaena + Gloeobacter + Thermobaculum + Thermogemmatispora + FFCH10602 + Dehalogenimonas + Dehalococcoides + Roseiflexus + Kouleothrix + Candidatus_Chlorothrix + Chloronema + Chloroflexus + Caldilinea + Ardenscatena + WCHB1-05 + T78 + SHD-231 + SHD-14 + Longilinea + C1_B004 + Anaerolinea + Ignavibacterium + Prosthecochloris + Waddlia + Candidatus_Rhabdochlamydia + Parachlamydia + Candidatus_Protochlamydia + LCP-26 + Caldithrix + Caldisericum + Fimbriimonas + Chthonomonas + CL0-1 + Armatimonas + Hydrogenobaculum + Candidatus_Chloracidobacterium + Bryobacter + Candidatus_Solibacter + Geothrix + Candidatus_Koribacter + Edaphobacter + Acidobacterium + Methanobrevibacter + Mucispirillum + Geovibrio + Flexistipes + Deferribacter + Elusimicrobium + Fibrobacteres-2 + Fibrobacter + Candidatus_Scalindua + Candidatus_Jettenia + Candidatus_Brocadia + Planctomyces + Pirellula + A17 + Isosphaera + Gemmata + Candidatus_Acetothermum + Thermodesulfovibrio + LCP-6 + HB118 + GOUTA19 + DCE29 + Candidatus_Magnetoovum + Candidatus_Magnetobacterium + BD2-6 + Leptospirillum + Nitrospira + JG37-AG-70 + Candidatus_Methylomirabilis + Lentisphaera + Gemmatimonas
32	Anoxybacillus + Anaerobacillus + Alkalibacillus
33	Bacillus
34	Geobacillus + Lentibacillus + Marinibacillus + Marinococcus + Natronobacillus + Oceanobacillus + Pontibacillus + Salimicrobium + Terribacillus + Thalassobacillus
35	Virgibacillus
36	Ammoniphilus + Aneurinibacillus + Brevibacillus + Cohnella
37	Paenibacillus
38	Lysinibacillus + Kurthia + Paenisporosarcina + Planococcus + Planomicrobium + Rummeliibacillus + Solibacillus + Sporosarcina + Ureibacillus + Viridibacillus + Pasteuria + Brochothrix + Kyrpidia + Alicyclobacillus + Sporolactobacillus + Pullulanibacillus + Thermicanus + Exiguobacterium + Thermoactinomyces + Shimazuella + Planifilum + Laceyella
39	Jeotgalicoccus + Macrococcus + Salinicoccus
40	Staphylococcus
41	Gemella + Turicibacter
42 - G2	Abiotrophia
43	Aerococcus
44	Alkalibacterium
45	Alloiococcus
46 - G2	Facklamia + Marinilactibacillus
47	Desemzia + Carnobacterium
48	Granulicatella
49	Trichococcus
50	Enterococcus
51	Tetragenococcus + Vagococcus
52	Lactobacillus + Pediococcus
53	Weissella + Leuconostoc + Fructobacillus
54	Lactococcus
55	Streptococcus
56 - G3	Clostridium + Clostridiisalibacter + Candidatus_Arthromitus + Caminicella + Caloranaerobacter + Caloramator + Caldanaerocella + Alkaliphilus + X02d06 + Geosporobacter_Thermotalea + Natronincola_Anaerovirgula + Oxobacter + Proteiniclasticum + SMB53 + Sarcina + Thermoanaerobacterium + Thermohalobacter + Tindallia_Anoxynatronum + Christensenella + Caldicoprobacter + Dehalobacterium
57	Pseudoramibacter_Eubacterium + Garciella + Anaerofustis + Alkalibacter + Acetobacterium + Lutispora + Gracilibacter + Heliorestis
58 - G4	Anaerostipes
59 - G4	Blautia
60 - G4	Butyrivibrio
61	Catonella
62 - G4	Coprococcus
63 - G4	Dorea + Epulopiscium + Lachnobacterium
64 - G4	Lachnospira
65	Moryella
66	Oribacterium + Pseudobutyrivibrio
67 - G4	Roseburia + Shuttleworthia
68 - G4	[Ruminococcus]
69	Desulfosporosinus + Desulfitobacter + Dehalobacter_Syntrophobotulus
70	Desulfotomaculum + Desulfotomaculum_Desulfovirgula + Desulfurispora + Niigata-25 + Pelotomaculum + Peptococcus + Sporotomaculum + WCHB1-84 + rc4-4
71 - G5	Faecalibacterium + Ethanoligenens + Anaerotruncus + Anaerofilum
72 - G5	Oscillospira
73	Ruminococcus
74	Anaerovibrio + Anaeromusa + Acidaminococcus + BSV43

ID	Child Buccal Microbiome: Genus level names
75	Dialister + G07 + Megamonas
76	Megasphaera + Mitsuokella + Pectinatus + Pelosinus
77	Phascolarctobacterium + Propionispora + Schwartzia
78	Selenomonas + Sporomusa + Succiniclasticum + Thermosinus
79	Veillonella + vadinHB04
80 - G3	WH1-8 + NP25 + Guggenheimella + Fusibacter + Acidaminobacter + Syntrophomonas + Symbiobacterium + YNPFFP6 + Sulfobacillus + Tepidibacter + Peptostreptococcus + Filifactor + Mogibacterium + Anaerovorax
81	Anaerococcus + X1_68 + Dethiosulfatibacter + Finegoldia + GW-34 + Gallicola + Helcococcus
82	Parvimonas + Peptoniphilus + Sedimentibacter + Sporanaerobacter + Tepidimicrobium + Tissierella_Soehngenia + WAL_1855D + ecb11 + ph2
83	Natroniella + Halanaerobacter + Acetohalobium + Halanaerobium + Natranaerobius + Candidatus_Contubernalis + KF-Gitt2-16 + Dethiobacter + Anaerobranca + A55_D21 + Thermodesulfobium + Coprothermobacter + Thermovenabulum + Thermoanaerobacter + Thermacetogenium + Moorella + Caldanaerobacter + Caldicellulosiruptor
84	Bulleidia + Allobaculum + Catenibacterium + Coprobacillus + Erysipelothrix + Holdemania + L7A_E11 + PSB-M-3 + RFN20 + Sharpea + [Eubacterium] + cc_115 + p-75-a5
85	Cetobacterium
86	Fusobacterium + Propionigenium + Psychrilyobacter + u114
87	Leptotrichia
88 - G6	Sneathia
89 - G6	Streptobacillus
90	Rhodoplanes + Rhodobium + Pedomicrobium + Parvibaculum + Hyphomicrobium + Devosia + Cohaesibacter + Pseudochrobactrum + Ochrobactrum + Nitrobacter + Bradyrhizobium + Bosea + Balneimonas + Afipia + Methylocella + Chelatococcus + Beijerinckia + Bartonella + Martelella + Fulvimarina + Aurantimonas + Methylobacterium + Pleomorphomonas + Methylosinus + Methylopila + Phyllobacterium + Nitratireductor + Mesorhizobium + Defluvibacter + Chelativorans + Aminobacter + Sinorhizobium + Shinella + Rhizobium + Kaistia + Candidatus_Liberibacter + Agrobacterium + Afifella + Xanthobacter + Labrys + Blastochloris + Azorhizobium + Thalassospira + Phenylobacterium + Mycoplana + Caulobacter + Brevundimonas + Asticcacaulis + Thalassobius + Shimia + Sagittula + Ruegeria + Rubellimicrobium + Rhodovulum + Rhodobacter + Rhodobaca + Phaeobacter + Paracoccus + Octadecabacter + Nautella + Marivita + Loktanella + Dinoroseobacter + Antarctobacter + Anaerospora + Amaricoccus + Oceanicaulis + Maricaulis + Hyphomonas + Hirschia
91	Telmatospirillum + Skermanella + Roseospira + Rhodovibrio + Rhodospirillum + Phaeospirillum + Novispirillum + Nisaea + Magnetospirillum + Inquilinus + Azospirillum + Swaminathania + Roseomonas + Roseococcus + Gluconobacter + Acidocella + Acidisphaera + Acidiphilium + Acetobacter + Wolbachia + Rickettsia + Neorickettsia + Ehrlichia + Candidatus_Neoehrlichia + Anaplasma + Zymomonas + Sphingopyxis + Sphingomonas + Sphingobium + Novosphingobium + Kaistobacter + Blastomonas + Lutibacterium + Erythrobacter
92 - G7	Achromobacter + Denitrobacter + Oligella + Pigmentiphaga
93 - G7	Sutterella + Tetrathiobacter
94	Burkholderia + Candidatus_Glomeribacter
95	Lautropia + Pandoraea + Salinispora
96 - G7	Aquabacterium + Alicycliphilus + Acidovorax + Comamonas + Curvibacter
97 - G7	Delftia + Diaphorobacter + Giesbergeria + Hydrogenophaga + Hylemonella + Lampropedia + Leptothrix
98 - G7	Limnobacter + Limnohabitans + Methylibium + Paucibacter + Pelomonas + Polaromonas + RS62 + Ramlibacter + Rhodoferax + Roseateles + Rubrivivax + Schlegelella + Simplicispira + Tepidimonas + Thiomonas + Variovorax + Verminephrobacter + Xenophilus
99 - G8	Janthinobacterium + Herminiimonas + Herbaspirillum + Cupriavidus + Collimonas
100	Oxalobacter + Polynucleobacter
101 - G8	Ralstonia
102	Conchiformibius + Chromobacterium + Aquitalea + Deefgea
103	Eikenella
104	Kingella + Microvirgula
105	Neisseria + Vitreoscilla + Vogesella
106	Nitrosovibrio + Nitrosospira + Methylotenera + Methylobacillus + Thiobacillus + Gallionella + Procabacter
107	TS34 + Sulfuritalea + Sterolibacterium + Rhodocyclus + Propionivibrio + Methyloversatilis + KD1-23 + K82 + Hydrogenophilus + Dok59 + Denitratisoma + Dechloromonas + Candidatus_Accumulibacter + C39 + Azovibrio + Azospira + Azoarcus + Thauera + Uliginosibacterium + Z-35 + Zoogloea + Thiobacter + Candidatus_Tremblaya
108	Lawsonia + Desulfovibrio + Bilophila + Desulfonatronum + Desulfomicrobium + Desulfovermiculus + Desulfonauticus + Desulfonatronovibrio + Nitrospina + Desulfotalea + Desulforhopalus + Desulfocapsa + Desulfobulbus + Desulfotignum + Desulfosarcina + Desulfofrigus + Desulfococcus + Desulfobacter + Desulfarculus + Bdellovibrio + Bacteriovorax + Desulfurella + Pelobacter + Geobacter + Desulfuromonas + Sorangium + Chondromyces + Plesiocystis + Myxococcus + Corallococcus + Anaeromyxobacter + Haliangium + Cystobacter + Syntrophobacter + Desulfacinum + Syntrophus + Smithella + Desulfomonile + Desulfobacca + Geothermobacterium + Candidatus_Entotheonella
109	Campylobacter + Arcobacter + Sulfurospirillum + Sulfurimonas + Sulfuricurvum + Helicobacter + Flexispira + Caminibacter + Mariprofundus
110 - G9	Succinivibrio + Ruminobacter + Anaerobiospirillum + Tolumonas + Oceanisphaera + Oceanimonas + Acidithiobacillus
111	HTCC2207 + HB2-32-21 + Glaciecola + Cellvibrio + Candidatus_Endobugula + BD2-13 + Alteromonas + Agarivorans + Marinimicrobium

ID	Child Buccal Microbiome: Genus level names
112	Marinobacter + Microbulbifer + ND137 + Spongiibacter + Umboniibacter + ZD0117 + nsmpVI18
113	Congregibacter + Moritella + Pseudidiomarina + Idiomarina + HTCC + Ferrimonas + Thalassomonas + Colwellia + Psychromonas + Shewanella + Rheinheimera + Alishewanella
114	Cardiobacterium
115 - G9	Thiovirga + Thiofaba + Halothiobacillus + Thiorhodospira + Thioalkalivibrio + Methylonatrum + Halorhodospira + Ectothiorhodospira + Thiorhodococcus + Thiocystis + Thiococcus + Thiocapsa + Nitrosococcus + Marichromatium + Halochromatium + Chromatium + Allochromatium + Methylomonas + Methylomicrobium + Methylocaldum + Crenothrix + Tatlockia + Legionella + Francisella + Rickettsiella + Coxiella + Aquicella
116 - G9	Candidatus_Blochmannia + Buchnera + Brenneria + Arsenophonus + Candidatus_Hamiltonella + Candidatus_Phlomobacter + Candidatus_Regiella + Citrobacter + Dickeya + Edwardsiella + Enterobacter
117 - G9	Erwinia + Gluconacetobacter
118 - G9	Klebsiella
119 - G9	Morganella + Photorhabdus + Plesiomonas + Proteus + Providencia
120 - G9	Salmonella
121 - G9	Serratia + Sodalis + Trabulsiella + Yersinia
122 - G9	Candidatus_Portiera + Chromohalobacter + Cobetia + Haererehalobacter
123 - G9	Halomonas + Kushneria
124 - G9	Oleispira + Oleibacter + Oceanospirillum + Nitrincola + Neptunomonas + Marinomonas + Marinobacterium + Amphritea + Hahella + Alcanivorax + Saccharospirillum + Reinekea + ML110J-20
125	Actinobacillus
126	Aggregatibacter + Avibacterium + Bibersteinia + Chelonobacter + Gallibacterium
127	Haemophilus + Mannheimia + Pasteurella
128	Acinetobacter + Alkanindiges
129 - G10	Enhydrobacter
130 - G10	Moraxella + Perlucidibaca
131 - G10	Psychrobacter
132	Pseudomonas + Azorhizophilus + Azomonas
133	Pseudoalteromonas
134	Vibrio + Salinivibrio + Photobacterium + Enterovibrio + Aliivibrio
135	Stenotrophomonas + Rhodanobacter + Pseudoxanthomonas + Lysobacter + Luteimonas + Luteibacter + Ignatzschineria + Dyella + Dokdonella + Arenimonas + Aquimonas + Thermomonas + Wohlfahrtiimonas + Xanthomonas + Xylella + Steroidobacter + Nevskia + Hydrocarboniphaga + Thiothrix + Thioploca + Leucothrix + E8 + Cocleimonas + CF-26 + Beggiatoa + B46 + Thiomicrospira + Thioalkalimicrobium + Piscirickettsia + Methylophaga + Thiohalorhabdus + Salinisphaera + Marinicella
136	E6 + vadinCA02 + Thermoanaerovibrio + Cloacibacillus + Candidatus_Tammella + TG5 + Pyramidobacter + PD-UASB-13 + HA73 + Aminobacterium + Anaerobaculum + Aminiphilus + SJA-88 + Turneriella + Leptospira + Leptonema + Brachyspira + Spironema + Borrelia + za29 + Treponema + Spirochaeta + X31d11 + Sphaerochaeta + so4B24 + ZA3312c + SargSea-WGS + SHAG537 + SGSH944 + Arctic95A-2 + Mycoplasma + Candidatus_Hepatoplasma + Mesoplasma + Entomoplasma + Asteroleplasma + Anaeroplasma + Candidatus_Phytoplasma + Acholeplasma + Thermotoga + Thermosipho + SC103 + S1 + Petrotoga + Kosmotoga + Geotoga + Fervidobacterium + AUTHM297
137	Akkermansia + Luteolibacter + Persicirhabdus + Prosthecobacter + Rubritalea + Verrucomicrobium + Pelagicoccus + Cerasicoccus + Puniceicoccus + MB11C04 + Coraliomargarita + Opitutus + LP2A + Candidatus_Methylacidiphilum + Pedosphaera + heteroC45_4W + OR-59 + Ellin506 + DA101 + Chthoniobacter + Candidatus_Xiphinematobacter + W5 + W22 + BHB21 + Vulcanithermus + Thermus + Meiothermus + Truepera + GBI-58 + B-42 + R18-435 + Deinococcus + Deinobacterium + CM44

Annex Table 1: List of the grouped abundances of the Child Buccal samples. All abundances have non negligible values in the sample. **G** indicates the group formed after the merging procedure to remove correlations.

ID	Child Stool Microbiome: Genus level names
1 - G1	Actinomyces + Actinobaculum + Arcanobacterium + Candidatus_Ancillula + Mobiluncus + No9 + Trueperella + Varibaculum + Actinopolyspora + Saccharothrix + Lentzea + Kibdelosporangium + Actinokineospora + Actinoalloteichus + Georgenia + Brevibacterium + Demequina + Cellulomonas + Actinotalea
2 - G1	Corynebacterium + Cryptosporangium + Dermabacter + Brachybacterium + Dermacoccus + Dermatophilus + Dietzia + Frankia + Modestobacter + Geodermatophilus + Blastococcus + Glycomyces + Gordonia + Tetrasphaera + Terracoccus + Serinicoccus + Phycicoccus + Kytococcus + Knoellia + Janibacter + Arsenicicoccus + Jonesia + Kineosporia + Kineococcus
3 - G1	Microbacterium + Leucobacter + Frigoribacterium + Curtobacterium + Cryocola + Cryobacterium + Clavibacter + Candidatus_Rhodoluna + Candidatus_Aquiluna + Agromyces + Agrococcus + Mycetocola + Pseudoclavibacter + Rathayibacter + Salinibacterium
4 - G2	Kocuria + Citricoccus + Arthrobacter + Microbispora
5 - G2	Micrococcus + Nesterenkonia + Renibacterium
6	Rothia + Sinomonas + Zhihengliuella
7 - G1	Rhodococcus + Nocardia + Mycobacterium + Virgisporangium + Verrucosispora + Solwaraspora + Pilimelia + Dactylosporangium + Couchioplanes + Catellatospora + Actinoplanes + Actinocatenispora + Propionicimonas + Pimelobacter + Nocardioides + Kribbella + Friedmanniella + Aeromicrobium + Actinopolymorpha + Streptomonospora + Prauseria + Nocardiopsis + Xylanimicrobium + Promicromonospora + Cellulosimicrobium
8 - G1	Propionibacterium + Microlunatus + Luteococcus + Tessaracoccus + Thermobispora + Saccharopolyspora + Saccharomonospora + Pseudonocardia + Prauserella + Jiangella + Amycolatopsis + Actinomycetospora + Rarobacter + Sanguibacter + Sporichthya
9 - G1	Streptomyces + Kitasatospora + Streptosporangium + Nonomuraea + Actinomadura + Actinocorallia + Actinoallomurus + Tsukamurella + Williamsia + Yaniella
10 - G3	Bifidobacterium + Alloscardovia + Bombiscardovia
11 - G3	Gardnerella + Scardovia
12	Adlercreutzia
13	Atopobium
14	Collinsella + Coriobacterium
15	Eggerthella + Slackia + Solirubrobacter + Patulibacter + Conexibacter + Rubrobacter + Nitriliruptor + Euzebya + Candidatus_Microthrix + Iamia + Ferrimicrobium
16	X5_7N15 + BF311
17	Bacteroides
18	Candidatus_Azobacteroides
19	Dysgonomonas
20	Paludibacter
21	Parabacteroides
22	Porphyromonas
23	Tannerella
24	Prevotella
25 - G4	AF12 + Blvii28
26 - G4	PW3
27 - G4	Rikenella
28	Odoribacter + Butyricimonas + BE24
29	Paraprevotella + CF231
30	YRC22
31	[Prevotella]
32	Capnocytophaga + Arenibacter + Aquimarina + Aequorivita + Cellulophaga
33	Flavobacterium + Gelidibacter + Gillisia + Gramella + Kordia + Lacinutrix + Leeuwenhoekiella + Lutimonas + Maribacter + Mesonia + Muricauda + Myroides + Polaribacter + Psychroflexus + Psychroserpens + Robiginitalea + Salegentibacter + Salinimicrobium + Sediminicola + Tenacibaculum + Ulvibacter + Winogradskyella + Zhouia + Zobellia
34	Elizabethkingia + Cloacibacterium + Chryseobacterium + Ornithobacterium + Riemerella + Wautersiella + Weeksella + Wandonia + Owenweeksia + Fluviicola + Cryomorpha + Crocinitomix + Brumimicrobium + Blattabacterium
35	Salinibacter + Rubricoccus + Rhodothermus + Salisaeta + KSA1 + Balneola + Sphingobacterium + Pedobacter + Olivibacter + Ucs1325 + SGUS912 + SC3-56 + Candidatus_Cardinium + Candidatus_Amoebophilus + Roseivirga + Reichenbachiella + Persicobacter + JTB248 + Fulvivirga + Flexithrix + Flexibacter + Flammeovirga + Sporocytophaga + Spirosoma + Runella + Rudanella + Rhodocytophaga + Pontibacter + Microscilla + Leadbetterella + Hymenobacter + Flectobacillus + Emticicia + Dyadobacter + Cytophaga + Adhaeribacter + Saprospira + Lewinella + Haliscomenobacter + Segetibacter + Sediminibacterium + Niabella + Flavisolibacter + Flavihumibacter + Chitinophaga

ID	Child Stool Microbiome: Genus level names
36	Synechococcus + Prochlorococcus + Thermosynechococcus + Acaryochloris + Pseudanabaena + Prochlorothrix + Leptolyngbya + Halomicronema + Arthronema + Symploca + Planktothrix + Planktothricoides + Phormidium + Oscillatoria + Microcoleus + Geitlerinema + Chroococcidiopsis + Spirulina + Prochloron + Microcystis + Cyanothece + Chroococcus + Rivularia + Calothrix + Nostoc + Gloeotrichia + Dolichospermum + Cylindrospermopsis + Anabaenopsis + Anabaena + Gloeobacter + Thermobaculum + Thermogemmatispora + FFCH10602 + Dehalogenimonas + Dehalococcoides + Roseiflexus + Kouleothrix + Candidatus_Chlorothrix + Chloronema + Chloroflexus + Caldilinea + Ardenscatena + WCHB1-05 + T78 + SHD-231 + SHD-14 + Longilinea + C1_B004 + Anaerolinea + Ignavibacterium + Prosthecochloris + Waddlia + Candidatus_Rhabdochlamydia + Parachlamydia + Candidatus_Protochlamydia + LCP-26 + Caldithrix + Caldisericum + Fimbriimonas + Chthonomonas + CL0-1 + Armatimonas + Hydrogenobaculum + Candidatus_Chloracidobacterium + Bryobacter + Candidatus_Solibacter + Geothrix + Candidatus_Koribacter + Edaphobacter + Acidobacterium + Methanobrevibacter + Mucispirillum + Geovibrio + Flexistipes + Deferribacter + Elusimicrobium + Fibrobacteres-2 + Fibrobacter
37 - G5	Anoxybacillus + Anaerobacillus + Alkalibacillus
38 - G5	Bacillus + Geobacillus + Lentibacillus + Marinibacillus + Marinococcus + Natronobacillus + Oceanobacillus + Pontibacillus + Salimicrobium + Terribacillus + Thalassobacillus + Virgibacillus
39 - G5	Paenibacillus + Cohnella + Brevibacillus + Aneurinibacillus + Ammoniphilus + Brochothrix + Kyrpidia + Alicyclobacillus + Pasteuria + Viridibacillus + Ureibacillus + Sporosarcina + Solibacillus + Rummeliibacillus + Planomicrobium + Planococcus + Paenisporosarcina + Lysinibacillus + Kurthia + Sporolactobacillus + Pullulanibacillus
40 - G5	Staphylococcus + Salinicoccus + Macrococcus + Jeotgalicoccus + Thermoactinomyces + Shimazuella + Planifilum + Laceyella + Exiguobacterium + Thermicanus
41 - G5	Gemella
42 - G6	Alloiococcus + Alkalibacterium + Aerococcus + Abiotrophia + Facklamia + Marinilactibacillus
43 - G6	Granulicatella + Desemzia + Carnobacterium + Trichococcus
44 - G6	Enterococcus + Tetragenococcus + Vagococcus
45	Lactobacillus + Pediococcus + Weissella + Leuconostoc + Fructobacillus
46	Lactococcus
47	Streptococcus
48	Turicibacter
49	Christensenella + Caldicoprobacter
50	Caloramator + Caldanaerocella + Alkaliphilus + X02d06 + Caloranaerobacter + Caminicella + Candidatus_Arthromitus
51	Clostridiisalibacter
52 - G7	Clostridium + Geosporobacter_Thermotalea + Natronincola_Anaerovirgula + Oxobacter
53	Proteiniclasticum + SMB53
54 - G7	Sarcina + Thermoanaerobacterium + Thermohalobacter + Tindallia_Anoxynatronum
55	Dehalobacterium
56	Anaerofustis + Alkalibacter + Acetobacterium + Garciella
57	Pseudoramibacter_Eubacterium
58	Gracilibacter + Lutispora + Heliorestis
59	Anaerostipes
60 - G8	Blautia
61	Butyrivibrio + Catonella
62 - G8	Coprococcus
63	Dorea
64	Epulopiscium
65	Lachnobacterium
66	Lachnospira
67 - G8	Moryella
68	Oribacterium
69 - G8	Pseudobutyrivibrio
70 - G8	Roseburia
71 - G8	Shuttleworthia
72	[Ruminococcus]
73	Desulfitobacter + Dehalobacter_Syntrophobotulus
74	Desulfosporosinus
75	Desulfotomaculum + Desulfotomaculum_Desulfovirgula + Desulfurispora + Niigata-25 + Pelotomaculum + Peptococcus + Sporotomaculum + WCHB1-84 + rc4-4
76	Peptostreptococcus + Filifactor + Tepidibacter
77	Anaerotruncus + Anaerofilum
78	Ethanoligenens
79	Faecalibacterium
80	Oscillospira
81	Ruminococcus
82	Symbiobacterium + YNPFFP6 + Sulfobacillus + Syntrophomonas
83 - G9	Acidaminococcus + Anaeromusa
84	Anaerovibrio + BSV43
85	Dialister + G07

ID	Child Stool Microbiome: Genus level names
86 - G9	Megamonas
87	Megasphaera + Mitsuokella + Pectinatus + Pelosinus
88	Phascolarctobacterium + Propionispora + Schwartzia
89	Selenomonas + Sporomusa
90	Succiniclasticum + Thermosinus
91	Veillonella + vadinHB04
92	Acidaminobacter
93	Fusibacter + Guggenheimella + NP25 + WH1-8
94	Mogibacterium + Anaerovorax + Thermodesulfobium + Coprothermobacter + Thermovenabulum + Thermoanaerobacter + Thermacetogenium + Moorella + Caldanaerobacter + Caldicellulosiruptor + Natranaerobius + Candidatus_Contubernalis + KF-Gitt2-16 + Dethiobacter + Anaerobranca + A55_D21 + Natroniella + Halanaerobacter + Acetohalobium + Halanaerobium
95 - G10	X1_68
96 - G10	Anaerococcus
97	Dethiosulfatibacter
98 - G10	Finegoldia + GW-34 + Gallicola
99	Helcococcus
100 - G10	Parvimonas
101	Peptoniphilus
102 - G10	Sedimentibacter + Sporanaerobacter + Tepidimicrobium + Tissierella_Soehngenia + WAL_1855D + ecb11 + ph2
103 - G11	Allobaculum
104 - G11	Bulleidia
105	Catenibacterium
106 - G11	Coprobacillus + Erysipelothrix
107	Holdemania + L7A_E11 + PSB-M-3
108 - G11	RFN20
109	Sharpea
110	[Eubacterium] + cc_115 + p-75-a5
111	Fusobacterium + Cetobacterium + Propionigenium + Psychrilyobacter + u114
112	Leptotrichia + Sneathia + Streptobacillus
113	Xanthobacter + Labrys + Blastochloris + Azorhizobium + Afifella + Sinorhizobium + Shinella + Rhizobium + Kaistia + Candidatus_Liberibacter + Agrobacterium + Phyllobacterium + Nitratireductor + Mesorhizobium + Defluvibacter + Chelativorans + Aminobacter + Pleomorphomonas + Methylosinus + Methylopila + Methylobacterium + Rhodoplanes + Rhodobium + Pedomicrobium + Parvibaculum + Hyphomicrobium + Devosia + Cohaesibacter + Pseudochrobactrum + Ochrobactrum + Nitrobacter + Bradyrhizobium + Bosea + Balneimonas + Afipia + Methylocella + Chelatococcus + Beijerinckia + Bartonella + Martelella + Fulvimarina + Aurantimonas + Thalassospira + Phenylobacterium + Mycoplana + Caulobacter + Brevundimonas + Asticcacaulis + Thalassobius + Shimia + Sagittula + Ruegeria + Rubellimicrobium + Rhodovulum + Rhodobacter + Rhodobaca + Phaeobacter + Paracoccus + Octadecabacter + Nautella + Marivita + Loktanella + Dinoroseobacter + Antarctobacter + Anaerospora + Amaricoccus + Oceanicaulis + Maricaulis + Hyphomonas + Hirschia + Telmatospirillum + Skermanella + Roseospira + Rhodovibrio + Rhodospirillum + Phaeospirillum + Novispirillum + Nisaea + Magnetospirillum + Inquilinus + Azospirillum + Swaminathania + Roseomonas + Roseococcus + Gluconobacter + Acidocella + Acidisphaera + Acidiphilium + Acetobacter + Wolbachia + Rickettsia + Neorickettsia + Ehrlichia + Candidatus_Neoehrlichia + Anaplasma + Zymomonas + Sphingopyxis + Sphingomonas + Sphingobium + Novosphingobium + Kaistobacter + Blastomonas + Lutibacterium + Erythrobacter
114	Sutterella + Pigmentiphaga + Oligella + Denitrobacter + Achromobacter + Tetrathiobacter
115	Lautropia + Candidatus_Glomeribacter + Burkholderia + Pandoraea + Salinispora
116	Comamonas + Aquabacterium + Alicycliphilus + Acidovorax + Curvibacter + Delftia + Diaphorobacter + Giesbergeria + Hydrogenophaga + Hylemonella + Lampropedia + Leptothrix + Limnobacter + Limnohabitans + Methylibium + Paucibacter + Pelomonas + Polaromonas + RS62 + Ramlibacter + Rhodoferax + Roseateles + Rubrivivax + Schlegelella + Simplicispira + Tepidimonas + Thiomonas + Variovorax + Verminephrobacter + Xenophilus
117	Oxalobacter + Janthinobacterium + Herminiimonas + Herbaspirillum + Cupriavidus + Collimonas + Polynucleobacter + Ralstonia
118	Neisseria + Microvirgula + Kingella + Eikenella + Deefgea + Conchiformibius + Chromobacterium + Aquitalea + Vitreoscilla + Vogesella + Methylotenera + Methylobacillus + Thiobacillus + Gallionella + Nitrosovibrio + Nitrosospira + Procabacter + Zoogloea + Z-35 + Uliginosibacterium + Thauera + TS34 + Sulfuritalea + Sterolibacterium + Rhodocyclus + Propionivibrio + Methyloversatilis + KD1-23 + K82 + Hydrogenophilus + Dok59 + Denitratisoma + Dechloromonas + Candidatus_Accumulibacter + C39 + Azovibrio + Azospira + Azoarcus + Thiobacter + Candidatus_Tremblaya
119	Bilophila + Desulfonatronum + Desulfomicrobium + Desulfovermiculus + Desulfonauticus + Desulfonatronovibrio
120	Desulfovibrio + Lawsonia
121	Pelobacter + Geobacter + Desulfuromonas + Desulfurella + Nitrospina + Desulfotalea + Desulforhopalus + Desulfocapsa + Desulfobulbus + Desulfotignum + Desulfosarcina + Desulfofrigus + Desulfococcus + Desulfobacter + Desulfarculus + Bdellovibrio + Bacteriovorax + Sorangium + Chondromyces + Plesiocystis + Myxococcus + Corallococcus + Anaeromyxobacter + Haliangium + Cystobacter + Syntrophobacter + Desulfacinum + Syntrophus + Smithella + Desulfomonile + Desulfobacca + Geothermobacterium + Candidatus_Entotheonella
122	Campylobacter + Arcobacter + Sulfurospirillum + Sulfurimonas + Sulfuricurvum + Helicobacter + Flexispira + Caminibacter + Mariprofundus

ID	Child Stool Microbiome: Genus level names
123 - G12	Shewanella + Psychromonas + Congregibacter + Moritella + Pseudidiomarina + Idiomarina + HTCC + Ferrimonas + Thalassomonas + Colwellia + nsmpVI18 + ZD0117 + Umboniibacter + Spongiibacter + ND137 + Microbulbifer + Marinobacter + Marinimicrobium + HTCC2207 + HB2-32-21 + Glaciecola + Cellvibrio + Candidatus_Endobugula + BD2-13 + Alteromonas + Agarivorans + Rheinheimera + Alishewanella + Succinivibrio + Ruminobacter + Anaerobiospirillum + Tolumonas + Oceanisphaera + Oceanimonas + Acidithiobacillus + Cardiobacterium + Thiovirga + Thiofaba + Halothiobacillus + Thiorhodospira + Thioalkalivibrio + Methylonatrum + Halorhodospira + Ectothiorhodospira + Thiorhodococcus + Thiocystis + Thiococcus + Thiocapsa + Nitrosococcus + Marichromatium + Halochromatium + Chromatium + Allochromatium
124 - G13	Candidatus_Blochmannia + Buchnera + Brenneria + Arsenophonus + Candidatus_Hamiltonella + Candidatus_Phlomobacter
125 - G13	Candidatus_Regiella
126 - G13	Citrobacter + Dickeya + Edwardsiella + Enterobacter
127 - G13	Erwinia
128 - G13	Gluconacetobacter
129 - G13	Klebsiella
130 - G13	Morganella
131 - G13	Photorhabdus
132 - G13	Plesiomonas
133	Proteus + Providencia
134 - G13	Salmonella
135 - G13	Serratia
136 - G13	Sodalis
137 - G13	Trabulsiella
138 - G13	Yersinia
139 - G12	Kushneria + Halomonas + Haererehalobacter + Cobetia + Chromohalobacter + Candidatus_Portiera + Hahella + Alcanivorax + Oleispira + Oleibacter + Oceanospirillum + Nitrincola + Neptunomonas + Marinomonas + Marinobacterium + Amphritea + Saccharospirillum + Reinekea + ML110J-20 + Methylomonas + Methylomicrobium + Methylocaldum + Crenothrix + Tatlockia + Legionella + Francisella + Rickettsiella + Coxiella + Aquicella
140	Actinobacillus
141 - G14	Aggregatibacter + Avibacterium + Bibersteinia + Chelonobacter + Gallibacterium
142 - G14	Haemophilus + Mannheimia + Pasteurella
143	Acinetobacter + Alkanindiges + Enhydrobacter
144	Moraxella + Perlucidibaca + Psychrobacter
145	Pseudomonas + Azorhizophilus + Azomonas
146 - G12	Vibrio + Salinivibrio + Photobacterium + Enterovibrio + Aliivibrio + Pseudoalteromonas + Thiothrix + Thioploca + Leucothrix + E8 + Cocleimonas + CF-26 + Beggiatoa + B46 + Thiomicrospira + Thioalkalimicrobium + Piscirickettsia + Methylophaga + Thiohalorhabdus + Salinisphaera
147 - G12	Stenotrophomonas + Rhodanobacter + Pseudoxanthomonas + Lysobacter + Luteimonas + Luteibacter + Ignatzschineria + Dyella + Dokdonella + Arenimonas + Aquimonas + Thermomonas + Wohlfahrtiimonas + Xanthomonas + Xylella + Steroidobacter + Nevskia + Hydrocarboniphaga + Marinicella
148	SJA-88 + Turneriella + Leptospira + Leptonema + Brachyspira + Spironema + Borrelia + za29 + Treponema + Spirochaeta + X31d11 + Sphaerochaeta + so4B24 + ZA3312c + SargSea-WGS + SHAG537 + SGSH944 + Arctic95A-2 + Candidatus_Scalindua + Candidatus_Jettenia + Candidatus_Brocadia + Planctomyces + Pirellula + A17 + Isosphaera + Gemmata + Candidatus_Acetothermum + Thermodesulfovibrio + LCP-6 + HB118 + GOUTA19 + DCE29 + Candidatus_Magnetoovum + Candidatus_Magnetobacterium + BD2-6 + Leptospirillum + Nitrospira + JG37-AG-70 + Candidatus_Methylomirabilis + Lentisphaera + Gemmatimonas + E6 + vadinCA02 + Thermoanaerovibrio + Cloacibacillus + Candidatus_Tammella + TG5 + Pyramidobacter + PD-UASB-13 + HA73 + Aminobacterium + Anaerobaculum + Aminiphilus + Mycoplasma + Candidatus_Hepatoplasma + Mesoplasma + Entomoplasma + Asteroleplasma + Anaeroplasma + Candidatus_Phytoplasma + Acholeplasma + Thermotoga + Thermosipho + SC103 + S1 + Petrotoga + Kosmotoga + Geotoga + Fervidobacterium + AUTHM297
149	Akkermansia + Luteolibacter + Persicirhabdus + Prosthecobacter + Rubritalea + Verrucomicrobium + Pelagicoccus + Cerasicoccus + Puniceicoccus + MB11C04 + Coraliomargarita + Opitutus + LP2A + Candidatus_Methylacidiphilum + Pedosphaera + heteroC45_4W + OR-59 + Ellin506 + DA101 + Chthoniobacter + Candidatus_Xiphinematobacter + W5 + W22 + BHB21 + Vulcanithermus + Thermus + Meiothermus + Truepera + GBI-58 + B-42 + R18-435 + Deinococcus + Deinobacterium + CM44

Annex Table 2: List of the grouped abundances of the Child Stool samples. All abundances have non negligible values in the sample. **G** indicates the group formed after the merging procedure to remove correlations.

ID	Mother Buccal Microbiome: Genus level names
1	Actinomyces + Actinobaculum + Arcanobacterium + Candidatus_Ancillula + Mobiluncus + N09 + Trueperella + Varibaculum + Actinopolyspora + Saccharothrix + Lentzea + Kibdelosporangium + Actinokineospora + Actinoalloteichus + Georgenia + Brevibacterium + Demequina + Cellulomonas + Actinotalea
2	Corynebacterium + Cryptosporangium + Dermabacter + Brachybacterium + Dermacoccus + Dermatophilus + Dietzia + Frankia + Modestobacter + Geodermatophilus + Blastococcus + Glycomyces
3	Gordonia + Tetrasphaera + Terracoccus + Serinicoccus + Phycicoccus + Kytococcus + Knoellia + Janibacter + Arsenicicoccus + Jonesia + Kineosporia + Kineococcus
4	Microbacterium + Leucobacter + Frigoribacterium + Curtobacterium + Cryocola + Cryobacterium + Clavibacter + Candidatus_Rhodoluna + Candidatus_Aquiluna + Agromyces + Agrococcus + Mycetocola + Pseudoclavibacter + Rathayibacter + Salinibacterium
5 - G1	Arthrobacter + Citricoccus
6	Kocuria + Microbispora
7 - G1	Micrococcus + Nesterenkonia + Renibacterium
8 - G1	Rothia + Sinomonas + Zhihengliuella
9	Verrucosispora + Solwaraspora + Pilimelia + Dactylosporangium + Couchioplanes + Catellatospora + Actinoplanes + Actinocatenispora + Virgisporangium + Mycobacterium
10	Rhodococcus + Nocardia + Propionicimonas + Pimelobacter + Nocardioides + Kribbella + Friedmanniella + Aeromicrobium + Actinopolymorpha + Streptomonospora + Prauseria + Nocardiopsis + Xylanimicrobium + Promicromonospora + Cellulosimicrobium
11	Propionibacterium + Microlunatus + Luteococcus + Tessaracoccus + Thermobispora + Saccharopolyspora + Saccharomonospora + Pseudonocardia + Prauserella + Jiangella + Amycolatopsis + Actinomycetospora + Rarobacter + Sanguibacter + Sporichthya
12	Streptomyces + Kitasatospora + Streptosporangium + Nonomuraea + Actinomadura + Actinocorallia + Actinoallomurus + Tsukamurella + Williamsia + Yaniella
13	Bifidobacterium + Alloscardovia + Bombiscardovia + Gardnerella
14	Scardovia
15	Atopobium + Adlercreutzia + Collinsella + Coriobacterium + Eggerthella + Slackia + Candidatus_Microthrix + Iamia + Ferrimicrobium + Nitriliruptor + Euzebya + Rubrobacter + Solirubrobacter + Patulibacter + Conexibacter
16	Bacteroides + BF311 + X5_7N15
17	Candidatus_Azobacteroides + Dysgonomonas
18 - G2	Paludibacter
19	Parabacteroides
20	Porphyromonas
21 - G2	Tannerella
22	Prevotella + Rikenella + PW3 + Blvii28 + AF12 + Odoribacter + Butyricimonas + BE24
23	YRC22 + Paraprevotella + CF231
24	[Prevotella]
25	Roseivirga + Reichenbachiella + Persicobacter + JTB248 + Fulvivirga + Flexithrix + Flexibacter + Flammeovirga + Sporocytophaga + Spirosoma + Runella + Rudanella + Rhodocytophaga + Pontibacter + Microscilla + Leadbetterella + Hymenobacter + Flectobacillus + Emticicia + Dyadobacter + Cytophaga + Adhaeribacter + Ucs1325 + SGUS912 + SC3-56 + Candidatus_Cardinium + Candidatus_Amoebophilus
26	Capnocytophaga + Arenibacter + Aquimarina + Aequorivita + Cellulophaga + Flavobacterium + Gelidibacter + Gillisia + Gramella + Kordia + Lacinutrix + Leeuwenhoekiella + Lutimonas + Maribacter + Mesonia + Muricauda + Myroides + Polaribacter + Psychroflexus + Psychroserpens + Robiginitalea + Salegentibacter + Salinimicrobium + Sediminicola + Tenacibaculum + Ulvibacter + Winogradskyella + Zhouia + Zobellia + Wandonia + Owenweeksia + Fluviicola + Cryomorpha + Crocinitomix + Brumimicrobium + Blattabacterium
27 - G3	Chryseobacterium
28 - G3	Cloacibacterium + Elizabethkingia + Ornithobacterium + Riemerella
29 - G3	Wautersiella + Weeksella
30	Salinibacter + Rubricoccus + Rhodothermus + Salisaeta + KSA1 + Balneola + Sphingobacterium + Pedobacter + Olivibacter + Saprospira + Lewinella + Haliscomenobacter + Segetibacter + Sediminibacterium + Niabella + Flavisolibacter + Flavihumibacter + Chitinophaga

ID	Mother Buccal Microbiome: Genus level names
31	Synechococcus + Prochlorococcus + Thermosynechococcus + Acaryochloris + Pseudanabaena + Prochlorothrix + Leptolyngbya + Halomicronema + Arthronema + Symploca + Planktothrix + Planktothricoides + Phormidium + Oscillatoria + Microcoleus + Geitlerinema + Chroococcidiopsis + Spirulina + Prochloron + Microcystis + Cyanothece + Chroococcus + Rivularia + Calothrix + Nostoc + Gloeotrichia + Dolichospermum + Cylindrospermopsis + Anabaenopsis + Anabaena + Gloeobacter + Thermobaculum + Thermogemmatispora + FFCH10602 + Dehalogenimonas + Dehalococcoides + Roseiflexus + Kouleothrix + Candidatus_Chlorothrix + Chloronema + Chloroflexus + Caldilinea + Ardenscatena + WCHB1-05 + T78 + SHD-231 + SHD-14 + Longilinea + C1_B004 + Anaerolinea + Ignavibacterium + Prosthecochloris + Waddlia + Candidatus_Rhabdochlamydia + Parachlamydia + Candidatus_Protochlamydia + LCP-26 + Caldithrix + Caldisericum + Fimbriimonas + Chthonomonas + CL0-1 + Armatimonas + Hydrogenobaculum + Candidatus_Chloracidobacterium + Bryobacter + Candidatus_Solibacter + Geothrix + Candidatus_Koribacter + Edaphobacter + Acidobacterium + Methanobrevibacter + Mucispirillum + Geovibrio + Flexistipes + Deferribacter + Elusimicrobium + Fibrobacteres-2 + Fibrobacter + Candidatus_Scalindua + Candidatus_Jettenia + Candidatus_Brocadia + Planctomyces + Pirellula + A17 + Isosphaera + Gemmata + Candidatus_Acetothermum + Thermodesulfovibrio + LCP-6 + HB118 + GOUTA19 + DCE29 + Candidatus_Magnetoovum + Candidatus_Magnetobacterium + BD2-6 + Leptospirillum + Nitrospira + JG37-AG-70 + Candidatus_Methylomirabilis + Lentisphaera + Gemmatimonas
32	Anoxybacillus + Anaerobacillus + Alkalibacillus
33 - G4	Bacillus
34 - G4	Geobacillus + Lentibacillus + Marinibacillus + Marinococcus + Natronobacillus + Oceanobacillus + Pontibacillus + Salimicrobium + Terribacillus + Thalassobacillus
35	Virgibacillus
36	Ammoniphilus + Aneurinibacillus + Brevibacillus + Cohnella
37	Paenibacillus
38	Lysinibacillus + Kurthia + Paenisporosarcina + Planococcus + Planomicrobium + Rummeliibacillus + Solibacillus + Sporosarcina + Ureibacillus + Viridibacillus + Pasteuria + Brochothrix + Kyrpidia + Alicyclobacillus + Sporolactobacillus + Pullulanibacillus + Thermicanus + Exiguobacterium + Thermoactinomyces + Shimazuella + Planifilum + Laceyella
39	Jeotgalicoccus + Macrococcus + Salinicoccus
40	Staphylococcus
41	Gemella + Turicibacter
42	Abiotrophia
43	Aerococcus
44	Alkalibacterium
45 - G5	Alloiococcus
46 - G5	Facklamia + Marinilactibacillus
47	Desemzia + Carnobacterium
48	Granulicatella
49	Trichococcus
50	Enterococcus
51	Tetragenococcus + Vagococcus
52	Lactobacillus + Pediococcus
53	Weissella + Leuconostoc + Fructobacillus
54	Lactococcus
55	Streptococcus
56 - G6	Clostridium + Clostridiisalibacter + Candidatus_Arthromitus + Caminicella + Caloranaerobacter + Caloramator + Caldanaerocella + Alkaliphilus + X02d06 + Geosporobacter_Thermotalea + Natronincola_Anaerovirgula + Oxobacter + Proteiniclasticum + SMB53 + Sarcina + Thermoanaerobacterium + Thermohalobacter + Tindallia_Anoxynatronum + Christensenella + Caldicoprobacter + Dehalobacterium
57	Alkalibacter + Acetobacterium + Anaerofustis + Garciella + Pseudoramibacter_Eubacterium + Lutispora + Gracilibacter + Heliorestis
58 - G7	Blautia + Anaerostipes
59	Butyrivibrio
60	Catonella
61 - G7	Coprococcus
62 - G7	Dorea + Epulopiscium + Lachnobacterium
63	Lachnospira
64	Moryella
65	Oribacterium + Pseudobutyrivibrio
66 - G7	Roseburia + Shuttleworthia
67 - G7	[Ruminococcus]
68	Desulfosporosinus + Desulfitobacter + Dehalobacter_Syntrophobotulus
69	Desulfotomaculum + Desulfotomaculum_Desulfovirgula + Desulfurispora + Niigata-25 + Pelotomaculum
70	Peptococcus + Sporotomaculum + WCHB1-84 + rc4-4
71	Filifactor
72	Peptostreptococcus + Tepidibacter
73	Faecalibacterium + Ethanoligenens + Anaerotruncus + Anaerofilum
74	Oscillospira
75	Ruminococcus

ID	Mother Buccal Microbiome: Genus level names
76	Anaerovibrio + Anaeromusa + Acidaminococcus + BSV43
77	Dialister + G07 + Megamonas
78	Megasphaera + Mitsuokella + Pectinatus + Pelosinus
79	Phascolarctobacterium + Propionispora
80	Schwartzia
81	Selenomonas + Sporomusa + Succiniclasticum + Thermosinus
82	Veillonella + vadinHB04
83 - G6	WH1-8 + NP25 + Guggenheimella + Fusibacter + Acidaminobacter + Syntrophomonas + Symbiobacterium + YNPFFP6 + Sulfobacillus
84	Mogibacterium + Anaerovorax
85	Anaerococcus + X1_68 + Dethiosulfatibacter + Finegoldia + GW-34 + Gallicola + Helcococcus
86	Parvimonas
87	Peptoniphilus + Sedimentibacter + Sporanaerobacter + Tepidimicrobium + Tissierella_Soehngenia + WAL_1855D + ecb11 + ph2
88	Natroniella + Halanaerobacter + Acetohalobium + Halanaerobium + Natranaerobius + Candidatus_Contubernalis + KF-Gitt2-16 + Dethiobacter + Anaerobranca + A55_D21 + Thermodesulfobium + Coprothermobacter + Thermovenabulum + Thermoanaerobacter + Thermacetogenium + Moorella + Caldanaerobacter + Caldicellulosiruptor
89	Bulleidia + Allobaculum + Catenibacterium + Coprobacillus + Erysipelothrix + Holdemania + L7A_E11 + PSB-M-3 + RFN20 + Sharpea + [Eubacterium] + cc_115 + p-75-a5
90	Cetobacterium
91	Fusobacterium + Propionigenium + Psychrilyobacter + u114
92	Leptotrichia + Sneathia + Streptobacillus
93	Rhodoplanes + Rhodobium + Pedomicrobium + Parvibaculum + Hyphomicrobium + Devosia + Cohaesibacter + Pseudochrobactrum + Ochrobactrum + Nitrobacter + Bradyrhizobium + Bosea + Balneimonas + Afipia + Methylocella + Chelatococcus + Beijerinckia + Bartonella + Martelella + Fulvimarina + Aurantimonas + Methylobacterium + Pleomorphomonas + Methylosinus + Methylopila + Phyllobacterium + Nitratireductor + Mesorhizobium + Defluvibacter + Chelativorans + Aminobacter + Sinorhizobium + Shinella + Rhizobium + Kaistia + Candidatus_Liberibacter + Agrobacterium + Afifella + Xanthobacter + Labrys + Blastochloris + Azorhizobium + Thalassospira + Phenylobacterium + Mycoplana + Caulobacter + Brevundimonas + Asticcacaulis + Thalassobius + Shimia + Sagittula + Ruegeria + Rubellimicrobium + Rhodovulum + Rhodobacter + Rhodobaca + Phaeobacter + Paracoccus + Octadecabacter + Nautella + Marivita + Loktanella + Dinoroseobacter + Antarctobacter + Anaerospora + Amaricoccus + Oceanicaulis + Maricaulis + Hyphomonas + Hirschia
94	Rhodovibrio + Rhodospirillum + Phaeospirillum + Novispirillum + Nisaea + Magnetospirillum + Inquilinus + Azospirillum + Roseospira + Skermanella + Telmatospirillum + Swaminathania + Roseomonas + Roseococcus + Gluconobacter + Acidocella + Acidisphaera + Acidiphilium + Acetobacter + Wolbachia + Rickettsia + Neorickettsia + Ehrlichia + Candidatus_Neoehrlichia + Anaplasma + Zymomonas + Sphingopyxis + Sphingomonas + Sphingobium + Novosphingobium + Kaistobacter + Blastomonas + Lutibacterium + Erythrobacter
95 - G8	Sutterella + Pigmentiphaga + Oligella + Denitrobacter + Achromobacter + Tetrathiobacter
96 - G8	Burkholderia + Candidatus_Glomeribacter
97 - G8	Lautropia + Pandoraea + Salinispora
98 - G8	Aquabacterium + Alicycliphilus + Acidovorax + Comamonas + Curvibacter + Delftia + Diaphorobacter + Giesbergeria + Hydrogenophaga + Hylemonella + Lampropedia + Leptothrix + Limnobacter + Limnohabitans + Methylibium + Paucibacter + Pelomonas + Polaromonas + RS62 + Ramlibacter + Rhodoferax + Roseateles + Rubrivivax + Schlegelella + Simplicispira + Tepidimonas + Thiomonas + Variovorax + Verminephrobacter + Xenophilus
99	Janthinobacterium + Herminiimonas + Herbaspirillum + Cupriavidus + Collimonas + Oxalobacter + Polynucleobacter
100	Ralstonia
101 - G9	Conchiformibius + Chromobacterium + Aquitalea + Deefgea
102	Eikenella
103	Kingella + Microvirgula
104 - G9	Neisseria + Vitreoscilla + Vogesella
105	Nitrosovibrio + Nitrosospira + Methylotenera + Methylobacillus + Thiobacillus + Gallionella + Procabacter
106	TS34 + Sulfuritalea + Sterolibacterium + Rhodocyclus + Propionivibrio + Methyloversatilis + KD1-23 + K82 + Hydrogenophilus + Dok59 + Denitratisoma + Dechloromonas + Candidatus_Accumulibacter + C39 + Azovibrio + Azospira + Azoarcus + Thauera + Uliginosibacterium + Z-35 + Zoogloea + Thiobacter + Candidatus_Tremblaya
107	Lawsonia + Desulfovibrio + Bilophila + Desulfonatronum + Desulfomicrobium + Desulfovermiculus + Desulfonauticus + Desulfonatronovibrio + Nitrospina + Desulfotalea + Desulforhopalus + Desulfocapsa + Desulfobulbus + Desulfotignum + Desulfosarcina + Desulfofrigus + Desulfococcus + Desulfobacter + Desulfarculus + Bdellovibrio + Bacteriovorax + Desulfurella + Pelobacter + Geobacter + Desulfuromonas + Sorangium + Chondromyces + Plesiocystis + Myxococcus + Corallococcus + Anaeromyxobacter + Haliangium + Cystobacter + Syntrophobacter + Desulfacinum + Syntrophus + Smithella + Desulfomonile + Desulfobacca + Geothermobacterium + Candidatus_Entotheonella
108	Campylobacter + Arcobacter + Sulfurospirillum + Sulfurimonas + Sulfuricurvum + Helicobacter + Flexispira + Caminibacter + Mariprofundus
109 - G10	Succinivibrio + Ruminobacter + Anaerobiospirillum + Tolumonas + Oceanisphaera + Oceanimonas + Acidithiobacillus
110 - G10	Marinobacter + Marinimicrobium + HTCC2207 + HB2-32-21 + Glaciecola + Cellvibrio + Candidatus_Endobugula + BD2-13 + Alteromonas + Agarivorans + Microbulbifer + ND137 + Spongiibacter + Umboniibacter + ZD0117 + nsmpVI18 + Thalassomonas + Colwellia + Ferrimonas + HTCC + Pseudidiomarina + Idiomarina + Moritella

ID	Mother Buccal Microbiome: Genus level names
111 - G10	Congregibacter + Psychromonas + Shewanella + Rheinheimera + Alishewanella
112	Cardiobacterium + Thiovirga + Thiofaba + Halothiobacillus + Thiorhodospira + Thioalkalivibrio + Methylonatrum + Halorhodospira + Ectothiorhodospira + Thiorhodococcus + Thiocystis + Thiococcus + Thiocapsa + Nitrosococcus + Marichromatium + Halochromatium + Chromatium + Allochromatium
113 - G10	Candidatus_Blochmannia + Buchnera + Brenneria + Arsenophonus + Candidatus_Hamiltonella + Candidatus_Phlomobacter
114 - G10	Candidatus_Regiella
115 - G10	Citrobacter + Dickeya + Edwardsiella + Enterobacter
116 - G10	Erwinia
117 - G10	Gluconacetobacter
118 - G10	Klebsiella
119 - G10	Morganella
120 - G10	Photorhabdus
121 - G10	Plesiomonas
122 - G10	Proteus + Providencia
123 - G10	Salmonella
124 - G10	Serratia
125 - G10	Sodalis
126 - G10	Trabulsiella
127 - G10	Yersinia
128 - G10	Methylomonas + Methylomicrobium + Methylocaldum + Crenothrix + Tatlockia + Legionella + Francisella + Rickettsiella + Coxiella + Aquicella
129 - G10	Cobetia + Chromohalobacter + Candidatus_Portiera + Haererehalobacter
130 - G10	Halomonas + Kushneria
131 - G10	Oleispira + Oleibacter + Oceanospirillum + Nitrincola + Neptunomonas + Marinomonas + Marinobacterium + Amphritea + Hahella + Alcanivorax + Saccharospirillum + Reinekea + ML110J-20
132	Actinobacillus
133	Aggregatibacter
134	Avibacterium + Bibersteinia
135	Chelonobacter + Gallibacterium
136	Haemophilus + Mannheimia + Pasteurella
137	Acinetobacter + Alkanindiges
138	Enhydrobacter
139	Moraxella + Perlucidibaca + Psychrobacter
140	Pseudomonas + Azorhizophilus + Azomonas
141 - G10	Pseudoalteromonas
142 - G10	Vibrio + Salinivibrio + Photobacterium + Enterovibrio + Aliivibrio
143 - G10	Stenotrophomonas + Rhodanobacter + Pseudoxanthomonas + Lysobacter + Luteimonas + Luteibacter + Ignatzschineria + Dyella + Dokdonella + Arenimonas + Aquimonas + Thermomonas + Wohlfahrtiimonas + Xanthomonas + Xylella + Steroidobacter + Nevskia + Hydrocarboniphaga + Thiothrix + Thioploca + Leucothrix + E8 + Cocleimonas + CF-26 + Beggiatoa + B46 + Thiomicrospira + Thioalkalimicrobium + Piscirickettsia + Methylophaga + Thiohalorhabdus + Salinisphaera + Marinicella
144 - G11	Treponema + Spirochaeta + X31d11 + za29 + Sphaerochaeta + Spironema + Borrelia + Brachyspira + SJA-88 + Turneriella + Leptospira + Leptonema + s04B24 + ZA3312c + SargSea-WGS + SHAG537 + SGSH944 + Arctic95A-2
145 - G11	TG5 + Pyramidobacter + PD-UASB-13 + HA73 + Aminobacterium + Anaerobaculum + Aminiphilus + vadinCA02 + Thermoanaerovibrio + Cloacibacillus + Candidatus_Tammella + E6
146	Mycoplasma + Candidatus_Hepatoplasma + Mesoplasma + Entomoplasma + Asteroleplasma + Anaeroplasma + Candidatus_Phytoplasma + Acholeplasma + Thermotoga + Thermosipho + SC103 + S1 + Petrotoga + Kosmotoga + Geotoga + Fervidobacterium + AUTHM297
147	Akkermansia + Luteolibacter + Persicirhabdus + Prosthecobacter + Rubritalea + Verrucomicrobium + Pelagicoccus + Cerasicoccus + Puniceicoccus + MB11C04 + Coraliomargarita + Opitutus + LP2A + Candidatus_Methylacidiphilum + Pedosphaera + heteroC45_4W + OR-59 + Ellin506 + DA101 + Chthoniobacter + Candidatus_Xiphinematobacter + W5 + W22 + BHB21 + Vulcanithermus + Thermus + Meiothermus + Truepera + GBI-58 + B-42 + R18-435 + Deinococcus + Deinobacterium + CM44

Annex Table 3: List of the grouped abundances of the Mother Buccal samples. All abundances have non negligible values in the sample. **G** indicates the group formed after the merging procedure to remove correlations.

BIBLIOGRAPHY

Afgan, Enis et al. (2016). "The Galaxy platform for accessible, reproducible and collaborative biomedical analyses: 2016 update". *Nucleic Acids Research* (cited on p. 69).

Albrecht, Matthew A. et al. (2015). "Longitudinal cognitive decline in the AIBL cohort: The role of APOE $\varepsilon 4$ status". *Neuropsychologia*, 75, pp. 411 –419 (cited on p. 58).

Andreasen, Aniels et al. (2001). "Evaluation of CSF-tau and CSF-Aβ42 as diagnostic markers for alzheimer disease in clinical practice". *Archives of Neurology*, 58(3), pp. 373–379 (cited on p. 8).

Arrieta, Marie-Claire et al. (2014). "The Intestinal Microbiome in Early Life: Health and Disease". *Frontiers in Immunology*, 5, p. 427 (cited on p. 125).

Bailey, Timothy L. (2011). "DREME: motif discovery in transcription factor ChIP-seq data". *Bioinformatics*, 27(12) (cited on p. 21).

Bailey, Timothy L. and Elkan, Charles (1994). "Fitting a mixture model by expectation maximization to discover motifs in biopolymers" (cited on p. 22).

Barber, Rina F., Reimherr, Matthew, and Schill, Thomas (2016). "The Function-on-Scalar LASSO with Applications to Longitudinal GWAS". *Under Revision* (cited on pp. 67, 72, 82, 96).

Barbu, Viorel and Precupanu, Teodor (2012). *Convexity and optimization in Banach spaces*. Springer Science & Business Media (cited on p. 77).

Barski, Artem and Zhao, Keji (2009). "Genomic Location Analysis by ChIP-Seq". *Journal of Cellular Biochemistry*, 107 (cited on p. 6).

Bauer, Daniel J. and Curran, Patrick J. (2003). "Distributional assumptions of growth mixture models: implications for overextraction of latent trajectory classes." *Psychological Methods*, 8(3), pp. 338–363 (cited on p. 58).

Bauschke, Heinz H. and Combettes, Patrick L. (2011). *Convex analysis and monotone operator theory in Hilbert spaces*. Springer Science & Business Media (cited on p. 77).

Berlinet, Alain and Thomas-Agnan, Christine (2011). *Reproducing kernel Hilbert spaces in probability and statistics*. Springer Science & Business Media (cited on pp. 74, 76, 77, 157).

Bernardi, Mara, Sangalli, Laura M., Secchi, Piercesare, and Vantini, Simone (2014). "Analysis of proteomics data: Block k-mean alignment". *Electron. J. Statist.* 8(2), pp. 1714–1723 (cited on pp. 15, 145).

Bernstein, Herbert J., Pople, John A., and Schneider, W. G. (1957). "The Analysis of Nuclear Magnetic Resonance Spectra". *Canadian Journal of Chemistry*, 35(1), pp. 67–83 (cited on p. 126).

Boor, Carl de (2001). *A Practical Guide to Splines*. Applied Mathematical Sciences. Springer New York (cited on p. 111).

Boyd, Stephen and Vandenberghe, Lieven (2004). *Convex optimization*. Cambridge university press (cited on p. 77).

Carey, Nessa (2012). *The Epigenetics Revolution: How Modern Biology is Rewriting Our Understanding of Genetics, Disease and Inheritance.* London: Icon Books Ltd (cited on p. 6).

Carrasquillo, Minerva M. et al. (2015). "Late-onset Alzheimer's risk variants in memory decline, incident mild cognitive impairment, and Alzheimer's disease". *Neurobiology of Aging*, 36(1), pp. 60–67 (cited on p. 58).

Chen, Yakuan, Goldsmith, Jeff, and Ogden, Todd R. (2016). "Variable selection in function-on-scalar regression". *Stat* (cited on pp. 67, 71, 111, 112).

Christensen, Helen et al. (2008). "The association of APOE genotype and cognitive decline in interaction with risk factors in a 65–69 year old community sample". *BMC Geriatrics*, 8(1), pp. 1–10 (cited on p. 58).

Chu, Wanghuan, Li, Runze, and Reimherr, Matthew (2016). "Feature screening for time-varying coefficient models with ultrahigh-dimensional longitudinal data". *The Annals of Applied Statistics*, 10(2), pp. 596–617 (cited on p. 92).

Clifford, R Jack Jr et al. (2013). "Tracking pathophysiological processes in Alzheimer's disease: an updated hypothetical model of dynamic biomarkers". *The Lancet Neurology*, 12(2), pp. 207 –216 (cited on pp. 46, 47).

Cremona, Marzia A. et al. (2015). "Peak shape clustering reveals biological insights". *BMC Bioinformatics*, 16(349) (cited on pp. 9, 24, 35, 37).

dbGaP (2009). *SHARP - National Heart, Lung, and Blood Institute SNP Health Association Asthma Resource Project*. `http://www.ncbi.nlm.nih.gov/projects/gap/cgi-bin/study.cgi?study_id=phs000166.v2.p1` (cited on p. 92).

Downer, Brian, Zanjani, Faika, and Fardo, David W. (2013). "The Relationship Between Midlife and Late Life Alcohol Consumption, APOEε4 and the Decline in Learning and Memory Among Older Adults". *Alcohol and Alcoholism*, 49(1), pp. 17–22 (cited on p. 58).

Dunford, Nelson and Schwartz, Jacob T. (1963). *Linear operators. Part 2: Spectral theory. Self adjoint operators in Hilbert space*. Interscience Publishers (cited on pp. 73, 156).

Edwards, Anthony W. F. (1963). "The Measure of Association in a 2×2 Table". *Journal of the Royal Statistical Society. Series A (General)*, 126(1), pp. 109–114 (cited on p. 55).

Ellis, Kathryn A et al. (2009). "The Australian Imaging, Biomarkers and Lifestyle (AIBL) study of aging: methodology and baseline characteristics of 1112 individuals recruited for a longitudinal study of Alzheimer's disease". *International Psychogeriatrics*, 21(4), pp. 672–687 (cited on p. 50).

Fan, Jonathan and Reimherr, Matthew (2016). "Adaptive Function-on-Scalar Regression". *arXiv preprint - arXiv:1610.07507* (cited on pp. 67, 72, 81, 82, 95).

Fan, Yingying, James, Gareth M., and Radchenko, Peter (2015). "Functional Additive Regression". *Annals of Statistics*, 43, pp. 2296–2325 (cited on p. 71).

Fisch, Robert O., Bilek, Mary K., and Ulstrom, Robert (1975). "Obesity and leanness at birth and their relationship to body habitus in later childhood". *Pediatrics*, 56 (cited on p. 109).

Folstein, Marshal F., Folstein, Susan E., and McHugh, Paul R. (1975). "Mini-mental state: A practical method for grading the cognitive state of patients for the clinician". *Journal of Psychiatric Research*, 12(3), pp. 189 –198 (cited on p. 7).

Fraley, Chris and Raftery, Adrian E. (2002). "Model-based Clustering, Discriminant Analysis and Density Estimation". *Journal of the American Statistical Association*, 97, pp. 611–631 (cited on p. 51).

Frisoni, Giovanni B. et al. (2007). "The topography of grey matter involvement in early and late onset Alzheimer's disease". *Brain*, 130(3), pp. 720–730 (cited on p. 51).

Gaffney, Scott and Smyth, Padhraic (2004). "Joint Probabilistic Curve Clustering and Alignment". *Advances in Neural Information Processing Systems NY. MIT Press* (cited on p. 46).

Gentleman, Robert C. et al. (2004). "Bioconductor: open software development for computational biology and bioinformatics." *Genome biology*, 5(10), R80–16 (cited on p. 11).

Gertheiss, Jan, Maity, Arnab, and Staicu, Ana-Maria (2013). "Variable selection in generalized functional linear models". *Stat*, 2(1), pp. 86–101 (cited on p. 71).

Gui, Hongsheng et al. (2014). "Influence of Alzheimer's disease genes on cognitive decline: the Guangzhou Biobank Cohort Study". *Neurobiology of Aging*, 35(10), 2422.e3 –2422.e8 (cited on p. 58).

Guo, Yuchun et al. (2010). "Discovering homotypic binding events at high spatial resolution". *Bioinformatics*, 26(24), pp. 3028–3034 (cited on p. 9).

Gupta, Shobhit, Stamatoyannopoulos, John A., Bailey, Timothy L., and Noble, William S. (2007). "Quantifying similarity between motifs". *Genome Biology*, 8(2), pp. 1–9 (cited on p. 22).

Gupta, Veer B. et al. (2015). "Follow-up plasma apolipoprotein E levels in the Australian Imaging, Biomarkers and Lifestyle Flagship Study of Ageing (AIBL) cohort". *Alzheimer's Research & Therapy*, 7(1), pp. 1–9 (cited on p. 51).

Hartstra, Annick V., Bouter, Kristien E.C., Bäckhed, Fredrik, and Nieuwdorp, Max (2014). "Insights Into the Role of the Microbiome in Obesity and Type 2 Diabetes". *Diabetes Care*, 38(1), pp. 159–165 (cited on p. 69).

Heltshe, James F. and Forrester, Nancy E. (1983). "Estimating Species Richness Using the Jackknife Procedure". *Biometrics*, 39(1), pp. 1–11 (cited on p. 113).

Hesketh, Kylie D. and Campbell, Karen J. (2010). "Interventions to prevent obesity in 0–5 year olds: an updated systematic review of the literature". *Obesity (Silver Spring)*, 18 (cited on p. 109).

Horváth, Lajos and Kokoszka, Piotr S. (2012). *Inference for Functional Data with Applications*. Springer (cited on pp. 79, 121).

Huang, Jian, Ma, Shuangge, and Zhang, Cun-Hui (2008). "Adaptive Lasso for sparse high-dimensional regression models". *Statistica Sinica*, pp. 1603–1618 (cited on p. 75).

Huang, Lei et al. (2016). *Refund*. `http://cran.r-project.org/web/packages/refund/` (cited on p. 112).

Huopaniemi, Ilkka et al. (2014). "Disease progression subtype discovery from longitudinal EMR data with a majority of missing values and unknown initial time points". *AMIA Annual Symposium Proceedings*, pp. 709–718 (cited on p. 47).

Jedynak, Bruno M. et al. (2012). "A computational neurodegenerative disease progression score: method and results with the Alzheimer's disease Neuroimaging Initiative cohort". *Neuroimage*, 63 (3). Alzheimer's Disease Neuroimaging Initiative, pp. 1478–1486 (cited on p. 46).

Jones, Harold E. and Bayley, Nancy (1941). "The Berkeley Growth Study". *Child Development*, 12(2), pp. 167–173 (cited on p. 5).

Kalliomäki, Marko, Carmen Collado, Maria, Salminen, Seppo, and Isolauri, Erika (2008). "Early differences in fecal microbiota composition in children may predict overweight". *The American Journal of Clinical Nutrition*, 87(3), pp. 534–538 (cited on p. 109).

Karatzoglou, Alexandros, Smola, Alex, Hornik, Kurt, and Zeileis, Achim (2004). "kernlab – An S4 Package for Kernel Methods in R". *Journal of Statistical Software*, 11(9), pp. 1–20 (cited on p. 157).

Kent, W James et al. (2002). "The Human Genome Browser at UCSC". *Genome Research*, 12 (6), pp. 996–1006 (cited on p. 11).

Kidder, Benjamin L., Hu, Gangqing, and Zhao, Keji (2011). "ChIP-Seq: technical considerations for obtaining high-quality data". *Nature Immunology*, 12 (10), pp. 918–922 (cited on p. 10).

Kiddle, Steven J. (2016). *Temporal Clustering*. `https://github.com/KHP-Informatics/TC`. Accessed: June 2016 (cited on p. 47).

Kiddle, Steven J. et al. (2010). "Temporal clustering by affinity propagation reveals transcriptional modules in Arabidopsis thaliana". *Bioinformatics*, 26 (3), pp. 1219–1233 (cited on p. 46).

Koleva, Petya T., Bridgman, Sarah L., and Kozyrskyj, Anita L. (2015). "The Infant Gut Microbiome: Evidence for Obesity Risk and Dietary Intervention". *Nutrients*, 7 (4), pp. 2237–2260 (cited on p. 109).

Laha, Radha G. and Rohatgi, Vijay K. (1979). *Probability Theory*. Wiley, New York (cited on p. 74).

Ley, Ruth, Turnbaugh, Peter J., Klein, Samuel J., and Gordon, Jeffrey I. (2007). "Microbial Ecology: Human gut microbes associated with obesity". *Nature*, 444 (7122), pp. 1022–1023 (cited on p. 109).

Lian, Heng (2013). "Shrinkage estimation and selection for multiple functional regression". *Statistica Sinica*, 23, pp. 51–74 (cited on p. 71).

Lim, Yen Ying et al. (2015). "APOEε4 moderates amyloid-related memory decline in preclinical Alzheimer's disease". *Neurobiology of Aging*, 36(3), pp. 1239 –1244 (cited on p. 58).

Love, Michael I., Huber, Wolfgang, and Anders, Simon (2014). "Moderated estimation of fold change and dispersion for RNA-seq data with DESeq2". *Genome Biology*, 15 (12), p. 550 (cited on p. 21).

Maria, Winnock et al. (2002). "Longitudinal analysis of the effect of apolipoprotein E ε4 and education on cognitive performance in elderly subjects: the PAQUID study". *Journal of Neurology, Neurosurgery, and Psychiatry*, 72(6), pp. 794–797 (cited on p. 58).

Marron, J. S., Ramsay, James O., Sangalli, Laura M., and Srivastava, Anuj (2015). "Functional Data Analysis of Amplitude and Phase Variation". *Statist. Sci.* 30(4), pp. 468–484 (cited on pp. 5, 16).

Matsui, Hidetoshi and Konishi, Sadanori (2011). "Variable selection for functional regression models via the L1 regularization". *Computational Statistics & Data Analysis*, 55(12), pp. 3304–3310 (cited on p. 71).

Mendoza-Parra, Marco Antonio, Nowicka, Malgorzata, Van Gool, Wouter, and Gronemeyer, Hinrich (2013). "Characterising ChIP-seq binding patterns by model-based peak shape deconvolution". *BMC Genomics*, 14(1), pp. 1–10 (cited on p. 9).

Monteiro, Paulo O. A. and Victora, Cesar G. (2005). "Rapid growth in infancy and childhood and obesity in later life: a systematic review". *Obesity Reviews*, 6(2), pp. 143–154 (cited on p. 69).

Mueller, Susanne G et al. (2005). "Ways toward an Early Diagnosis in Alzheimer's Disease: The Alzheimer's Disease Neuroimaging Initiative (ADNI)." *Alzheimer's & Dementia*, pp. 55–66 (cited on pp. 46, 50).

Neville, Jon et al. (2015). "Development of a unified clinical trial database for Alzheimer's disease". *Alzheimer's & Dementia*, 11(10), pp. 1212 –1221 (cited on pp. 46, 50).

Ogden, Cynthia L., Carroll, Margaret D., Kit, Brian K., and Flegal, Katherine M. (2014). "Prevalence of childhood and adult obesity in the United States, 2011–2012". *JAMA*, 311 (cited on p. 109).

Pagès, Hervé, Aboyoun, Patrick, Gentleman, Robert, and DebRoy, Saikat (2016). *Biostrings: String objects representing biological sequences, and matching algorithms*. R package version 2.40.2 (cited on p. 21).

Parodi, Alice C.L. et al. (2015). *fdakma: Functional Data Analysis: K-Mean Alignment*. R package version 1.2 (cited on p. 144).

Parodi, Alice C.L. et al. (2016). *FunChIP: Clustering and Alignment of ChIP-Seq peaks based on their shapes*. R package version 1.0.0 (cited on pp. 7, 11).

Patriarca, Mirco, Sangalli, Laura M., Secchi, Piercesare, and Vantini, Simone (2014). "Analysis of spike train data: An application of k-mean alignment". *Electron. J. Statist.* 8(2), pp. 1769–1775 (cited on pp. 15, 145).

Paul, Ian M. et al. (2014). "The Intervention Nurses Start Infants Growing on Healthy Trajectories (INSIGHT) study". *BMC Pediatrics*, 14(1), pp. 1–15 (cited on p. 109).

Perera, Gayan et al. (2014). "Factors Associated with Response to Acetylcholinesterase Inhibition in Dementia: A Cohort Study from a Secondary Mental Health Care Case Register in London". *PLoS ONE*, 9 (11) (cited on p. 59).

Proust-Lima, Cècile, Amieva, Hlne, Dartigues, Jean-Franois, and Jacqmin-Gadda, Hlne (2007). "Sensitivity of Four Psychometric Tests to Measure Cognitive Changes in Brain Aging-Population based Studies". *American Journal of Epidemiology*, 165(3), pp. 344–350 (cited on p. 59).

Proust-Lima, Cècile, Philipps, Viviane, and Liquet, Benoit (2015). "Estimation of extended mixed models using latent classes and latent processes: the R package lcmm". *ArXiV* (cited on p. 45).

Rajan, Kumar B., Skarupski, Kimberly A., Rasmussen, Heather E., and Evans, Denis A. (2014). "Gene-Environment Interaction of Body Mass Index and Apolipoprotein Eε4 Allele on Cognitive Decline". *Alzheimer disease and associated disorders*, 28(2), pp. 134–140 (cited on p. 58).

Rajeevan, Haseena et al. (1999). *ALFRED - The ALlele FREquency Database*. `https://alfred.med.yale.edu/alfred/AboutALFRED.asp`. Yale University (cited on p. 95).

Ramsay, James O. and Silverman, Bernard W. (2005). *Functional Data Analysis*. 2nd. New York: Springer-Verlag (cited on pp. 5, 14, 16, 67, 111, 121, 122, 142).

Ramsay, James O., Wickham, Hadley, Graves, Spencer, and Hooker, Giles (2014). *fda: Functional Data Analysis*. R package version 2.4.4 (cited on p. 142).

Rand, William M. (1971). "Objective Criteria for the Evaluation of Clustering Methods". *Journal of the American Statistical Association*, 66(336), pp. 846–850 (cited on p. 51).

Repapi, Emmanouela et al. (2010). "Genome-wide association study identifies five loci associated with lung function". *Nature genetics*, 42(1), pp. 36–44 (cited on p. 94).

Rhee, Ho Sung and Pugh, B. Franklin (2012). "ChIP-exo Method for Identifying Genomic Location of DNA-Binding Proteins with Near-Single-Nucleotide Accuracy". *Current protocols in molecular biology / edited by Frederick M. Ausubel ... [et al.]* 21 (cited on p. 7).

Sabò, Arianna et al. (2014). "Selective transcriptional regulation by Myc in cellular growth control and lymphomagenesis". *Nature*, 511, pp. 488–492 (cited on pp. 13, 20).

Sakoe, Hiroaki and Chiba, Seibi (1978). "Dynamic programming algorithm optimization for spoken word recognition". *IEEE Transactions on Acoustics, Speech, and Signal Processing*, 26, pp. 43–49 (cited on p. 5).

Sangalli, Laura M., Secchi, Piercesare, Vantini, Simone, and Vitelli, Valeria (2010). "k-mean alignment for curve clustering". *Computational Statistics & Data Analysis*, 54(5), pp. 1219 –1233 (cited on pp. 5, 15, 16, 46, 49, 144).

Sangalli, Laura M., Secchi, Piercesare, and Vantini, Simone (2014). "Analysis of AneuRisk65 data: k-mean alignment". *Electronic Journal of Statistics*, 8(2), pp. 1891–1904 (cited on pp. 15, 145).

Shor, Naum Z. (2012). *Minimization methods for non-differentiable functions*. Vol. 3. Springer Science & Business Media (cited on p. 77).

Simpson, Edward H. (1949). "Measurement of diversity". *Nature*, 163(688) (cited on p. 113).

Srivastava, Anuj and Klassen, Eric P. (2016). *Functional and Shape Data Analysis*. Springer Series in Statistics (cited on p. 5).

Strachan, David P. et al. (2007). "Lifecourse influences on health among British adults: effects of region of residence in childhood and adulthood". *International journal of epidemiology*, 36(3), pp. 522–531 (cited on p. 94).

Tanzi, Rudolph E. and Bertram, Lars (2001). "New Frontiers in Alzheimer's Disease Genetics". *Neuron*, 32(2), pp. 181 –184 (cited on p. 8).

The Childhood Asthma Management Program Research Group (1999). "The Childhood Asthma Management Program (CAMP): design, rationale, and methods". *Controlled Clinical Trials*, 20, pp. 91–120 (cited on p. 92).

Tombaugh, Tom N. and McIntyre, Nancy J. (1992). "The Mini-Mental State Examination: A Comprehensive Review". *Journal of the American Geriatrics Society*, 40(9), pp. 922–935 (cited on p. 51).

Tonelli, Claudia et al. (2015). "Genome-wide analysis of p53 transcriptional programs in B cells upon exposure to genotoxic stress in vivo". *Oncotarget*, 6(28) (cited on p. 20).

Turnbaugh, Peter J. et al. (2006). "An obesity-associated gut microbiome with increased capacity for energy harvest". *Nature*, 444 (7122) (cited on p. 124).

Ursell, Luke K et al. (2012). "The interpersonal and intrapersonal diversity of human-associated microbiota in key body sites". *The Journal of allergy and clinical immunology*, 129(5), pp. 1204–1208 (cited on p. 113).

Vantini, Simone (2012). "On the definition of phase and amplitude variability in functional data analysis". *TEST*, 21(4), pp. 676–696 (cited on pp. 5, 16).

Weuve, Jennifer et al. (2015). "Guidelines for reporting methodological challenges and evaluating potential bias in dementia research". *Alzheimer's & Dementia*, 11(9), pp. 1098 –1109 (cited on p. 58).

Whitaker, Robert C. et al. (1997). "Predicting obesity in young adulthood from childhood and parental obesity". *The New England Journal of Medicine*, 337 (cited on p. 109).

W.H.O. (1948). *World Health Organization* (cited on p. 110).

Wilbanks, Elizabeth G. and Facciotti, Marc T. (2010). "Evaluation of Algorithm Performance in ChIP-Seq Peak Detection". *PLoS ONE*, 5(7), pp. 1–12 (cited on pp. 9, 10).

Wilkosz, Patricia A. et al. (2009). "Trajectories of cognitive decline in Alzheimer's disease". *International Psychogeriatrics*, 22, pp. 281–290 (cited on pp. 45, 46, 53).

Yang, Eric et al. (2011). "Quantifying the pathophysiological timeline of Alzheimer's disease". *Journal of Alzheimer's Disease*, 26, pp. 745–753 (cited on p. 46).

Yao, Fang, Mller, Hans-Georg, and Wang, Jane-Ling (2005). "Functional Data Analysis for Sparse Longitudinal Data". *Journal of the American Statistical Association*, 100(470), pp. 577–590 (cited on pp. 111, 112).

Zhang, Cun-Hui (2010). "Nearly unbiased variable selection under minimax concave penalty". *The Annals of statistics*, pp. 894–942 (cited on p. 72).

Zhang, Xuekui et al. (2011). "PICS: Probabilistic Inference for ChIP-seq". *Biometrics*, 67(1), pp. 151–163 (cited on p. 9).

Zhang, Yong et al. (2008). "Model-based Analysis of ChIP-Seq (MACS)". *Genome Biology*, 9(9), pp. 1–9 (cited on pp. 6, 21).

Zhao, Peng and Yu, Bin (2006). "On model selection consistency of Lasso". *Journal of Machine Learning Research*, 7(Nov), pp. 2541–2563 (cited on p. 82).