PUBLISHED JOURNAL ARTICLE
Cremona, Chiaromonte (2023) Probabilistic K-means with local alignment for clustering and motif discovery in functional data. Journal of Computational and Graphical Statistics.
https://doi.org/10.1080/10618600.2022.2156522

Probabilistic K-means with local alignment for clustering and motif discovery in functional data

Marzia A. Cremona *
Dept. of Operations and Decision Systems, Université Laval
and
Francesca Chiaromonte
Dept. of Statistics, The Pennsylvania State University

November 30, 2022

Abstract

We develop a new method to locally cluster curves and discover functional motifs, i.e. typical shapes that may recur several times along and across the curves capturing important local characteristics. In order to identify these shared curve portions, our method leverages ideas from functional data analysis (joint clustering and alignment of curves), bioinformatics (local alignment through the extension of high similarity seeds) and fuzzy clustering (curves belonging to more than one cluster, if they contain more than one typical shape). It can employ various dissimilarity measures and incorporate derivatives in the discovery process, thus exploiting complex facets of shapes. We demonstrate the performance of our method with an extensive simulation study, and show how it generalizes other clustering methods for functional data. Finally, we provide real data applications to Italian Covid-19 death curves and Omics data related to mutagenesis.

Keywords: Clustering; Functional data analysis; Local alignment; Motif discovery

*M.A. Cremona is also affiliated to CHU de Québec – Université Laval Research Center and Dept. of Statistics, Penn State University. F. Chiaromonte is also affiliated to the Inst. of Economics and EMbeDS, Sant'Anna School of Advanced Studies. This work was partially funded by the Eberly College of Science, the Institute for Cyberscience and the Huck Institutes of the Life Sciences (Penn State University); NSF award DMS-1407639; and Tobacco Settlement and CURE funds of the PA Department of Health. M.A. Cremona acknowledge the support of the NSERC. We thank Matthew Reimherr and Piercesare Secchi for discussions about functional data methodology; Kateryna D. Makova and Di (Bruce) Chen for help with the mutagenesis application; Valeria Vitelli and Davide Floriello for their sparse functional clustering code.

1 Introduction

Given a set of curves, we consider the problem of discovering *functional motifs* inside them, i.e. typical shapes (continuous curve portions) that may recur within each curve, and across several curves in the set (see Fig. 1). Some of these motifs may be present in most of the curves, but in different positions. Conversely, other motifs may characterize subgroups of curves and thus differentiate among them based on local shape similarities. We provide a novel method for functional motif discovery that aligns curves locally to identify their shared continuous portions, employing different definitions of (dis)similarity. Importantly, neither the motifs nor their number, lengths, or radii (i.e., the maximum dissimilarity between a motif shape and a portion of curve containing it) need to be known in advance; lengths and radii are specific to each motif.

Our motivating example is the study of mutagenesis, i.e. the processes that generate mutations in the DNA sequence of an organism. The aim is to identify "signature shapes" in multidimensional curves consisting of multiple types of mutation rates measured at high resolution along the genome (see Subsection 5.2). Each of these shapes, or motifs, represents a specific mutagenesis pattern, which recurs in a set of genomic regions and

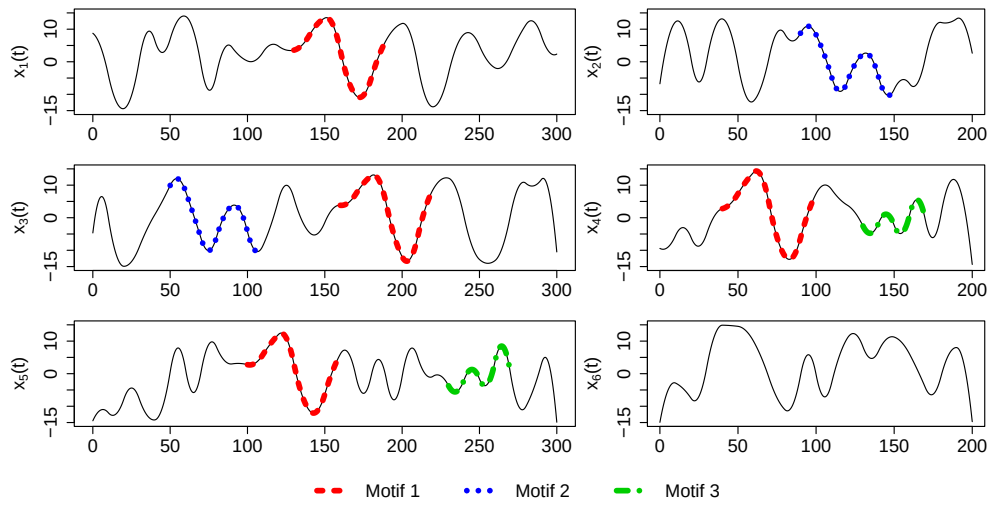


Figure 1: Example of curves comprising three functional motifs (dashed, dotted and dot-dash continuous portions of curves, respectively; one curve has no motifs).

is characterized by a certain genomic landscape. Functional motif discovery can also be useful to study protein-DNA interactions – an area in which shape has already been shown to carry biological information (Cremona et al., 2015). For instance, functional motifs in the 1-dimensional curves of ChIP-seq signals could help us distinguish different protein-DNA binding. Functional motifs are relevant in many other domains. For example, in finance, one may be interested in detecting recurrent patterns in the time series of asset prices related to different companies. This is an important problem in technical trading analysis, whose rules aim at predicting price changes based on observed patterns. A related problem is the detection of financial bubbles – which are characterized by a rapid surge in prices, not justified by the fundamentals, followed by an unexpected crash – in data such as stock market indices or exchange rates. Outside the "Omics" and financial fields, detecting functional motifs in weather time series may help us understand some aspects of climate change, while detecting functional motifs in time series generated by wearable devices, e.g., accelerometer data, may help us characterize patterns of physical activity.

During the last two decades, the analysis of curves has received increasing attention and interest in the statistical literature. Indeed, several functional data analysis methods have been developed and applied in many fields (see, e.g., Ramsay and Silverman, 2005; Ferraty and Vieu, 2006; Horváth and Kokoszka, 2012). Several algorithms have been proposed to cluster aligned functional data (reviewed in Jacques and Preda, 2014). Since functional data are very often misaligned, algorithms have also been proposed to simultaneously cluster and align curves (Liu and Yang, 2009; Sangalli et al., 2010; Park and Ahn, 2017). All these methods consider the curves globally, over their entire domain of definition. However, in many applications, separation in groups may occur only on a portion of the domain; this type of clustering structure might be missed by methods that consider curves in their entirety. The multivariate counterpart of this domain selection problem is usually referred to as feature selection and has been widely studied (see, e.g., Friedman and Meulman, 2004; Witten and Tibshirani, 2010). In the functional framework, Fraiman et al. (2016) and Floriello and Vitelli (2017) proposed methods to cluster curves while performing feature (i.e. domain) selection. More recently, Vitelli (2019) integrated curve alignment in the

sparse clustering procedure.

The problem of *functional motif discovery* we tackle here is more general and, to the best of our knowledge, it has never been studied in the statistical literature. To identify motifs, we define clusters locally on continuous portions of the misaligned curves and allow each cluster to contain multiple portions of the same curve (i.e. multiple instances of the same functional motif). In addition, we allow each curve to belong to zero, one, or multiple clusters (i.e. to comprise zero, one, or multiple functional motifs). This problem is the continuous version of sequence motif discovery, which is ubiquitous in bioinformatics and "Omics" sciences (see, e.g., Bailey et al., 2006) and consists of searching for highly similar patterns in a set of DNA or protein sequences. While these are discrete sequences of symbols (4 nucleotides, or 20 amino acids), we consider curves that can attain any real values and can be multivariate (i.e. take values in $\mathbb{R}^d$). A similar problem for time series has been addressed by the data mining community (Lin et al., 2002; Mueen et al., 2009; Yeh et al., 2016, 2018) defining a motif as a pattern repeated multiple times within a single time series. Available tools generally employ the Euclidean distance or the correlation between portions of the time series. They usually require as input the length and the number of motifs to be found, although Linardi et al. (2018) recently introduced an algorithm that finds all motifs in a given range of lengths. Importantly, these tools require a user-specified minimum distance within which two portions of the time series are considered the same motif (i.e. a motif radius), and this distance is the same across motifs.

We embed the problem of functional motif discovery in a full-blown functional framework, which allows us to capture complex shape characteristics by incorporating derivatives in the discovery process. The functional framework also allows us to rigorously define variability within each motif, and to naturally reduce noise in the curves through smoothing. Our novel method, *probabilistic K-means with local alignment* (probKMA), leverages ideas from functional data analysis, bioinformatics, and fuzzy clustering in order to identify K shared curve portions, which represent K candidate functional motifs in the set of curves under consideration. Similar to the K-means with (global) alignment of Sangalli et al. (2010), we simultaneously perform clustering and alignment of curves. However, we employ

local alignment in place of their global alignment. Also, similar to BLAST-type algorithms in bioinformatics (Altschul et al., 1990), we perform local alignments through the extension of high similarity seeds. Finally, similar to fuzzy clustering in which points can belong to multiple clusters (Bezdek, 1981; Bezdek et al., 1984), curves can be associated with zero, one, or more than one cluster (if they contain zero, one, or more than one typical shape).

The article is organized as follows. In Section 2 we present probKMA's theoretical setting, formulate it as an optimization problem, derive necessary conditions for its solution, and describe its algorithmic implementation. In Section 3 we discuss the evaluation of the clusters produced and identification of the motifs discovered. In Section 4 we provide simulation studies to evaluate probKMA and compare it to other approaches. We present real data applications in Section 5 and provide concluding remarks in Section 6.

2 Probabilistic K-means with local alignment

2.1 Optimization problem and necessary conditions

We consider a set of N (d-dimensional) curves $\mathbf{x}_i : \mathbb{R} \longrightarrow \mathbb{R}^d$, $i = 1, \ldots, N$. Our goal is to identify K (d-dimensional) cluster centers $\mathbf{v}_k$ – representing K candidate motifs – to which the curves are, locally, highly similar with respect to a distance $d(\cdot, \cdot)$. Without loss of generality, we can assume that the domain of each $\mathbf{v}_k$ starts at 0; in symbols, $\mathbf{v}_k : (0, c_k) \longrightarrow \mathbb{R}^d$, $k = 1, \ldots, K$, with unknown lengths $c_1, \ldots, c_K \in [c_{min}, c_{max}]$. Then, each curve is aligned to each cluster center $\mathbf{v}_k$ as to minimize their distance in the interval $(0, c_k)$. Alignment is performed composing each curve $\mathbf{x}_i$ with a warping function $h_{k,i} : \mathbb{R} \longrightarrow \mathbb{R}$ from a class W so that the curve portion matching the cluster center moves to the interval $(0, c_k)$. Here we consider shifts $W = \{h : t \longmapsto t + s; s \in \mathbb{R}\}$, but our method can be generalized to other warping functions commonly employed in the functional data analysis literature (see, e.g., Chapter 7 of Ramsay and Silverman, 2005). Because of the focus on local similarity, a curve can belong to more than one cluster; that is, different portions of a curve can be similar to portions of other curves. Hence, mimicking fuzzy clustering (see, e.g., Bezdek, 1981; Bezdek et al., 1984), we assign to each curve $\mathbf{x}_i$ a probability $p_{k,i}$ to be a member

of each cluster k. We define a membership function $p_k : \{\mathbf{x}_1, \ldots, \mathbf{x}_N\} \longrightarrow [0,1]$ for each $k = 1, \ldots, K$, with $p_k(\mathbf{x}_i) = p_{k,i}$, requiring that $\sum_{k=1}^{K} p_{k,i} = 1$ for all $i = 1, \ldots, N$, and that $\sum_{i=1}^{N} p_{k,i} > 0$ for all $k = 1, \ldots, K$. Each $p_{k,i}$ corresponds to a particular shift $s_{k,i}$ of the curve $\mathbf{x}_i$; namely, the one that minimizes the distance between $\mathbf{x}_i$ and $\mathbf{v}_k$ given all constraints. We denote $\mathbf{S} = [s_{k,i}] \in \mathbb{R}^{K \times N}$ and $\mathbf{P} = [p_{k,i}] \in [0,1]^{K \times N}$, where $\mathbb{R}^{K \times N}$ and $[0,1]^{K \times N}$ indicate the space of matrices of dimension $K \times N$ with elements in $\mathbb{R}$ and $[0,1]$, respectively.

Consider the cluster center lengths $c_1, \ldots, c_K$ as fixed (identification of $c_k \in [c_{min}, c_{max}]$ is discussed in the next Subsection). ProbKMA can be formulated as the following optimization problem: find K cluster centers $\mathbf{v}_1, \ldots, \mathbf{v}_K$, membership probabilities $\mathbf{P}$ and shifts $\mathbf{S}$ that minimize the generalized least-squares functional

$$J_m(\mathbf{P}, \mathbf{S}, \mathbf{v}_1, \ldots, \mathbf{v}_K) = \sum_{i=1}^{N} \sum_{k=1}^{K} (p_{k,i})^m \, d^2\left(\tilde{\mathbf{x}}_{i,s_{k,i}}, \mathbf{v}_k\right) \tag{1}$$

under the constraints $p_{k,i} \in [0,1]$, $\forall i, k$; $\sum_{k=1}^{K} p_{k,i} = 1$, $\forall i$; and $\sum_{i=1}^{N} p_{k,i} > 0$, $\forall k$. Here $m > 1$ is a fixed parameter controlling the degree of fuzziness, and $\tilde{\mathbf{x}}_{i,s_{k,i}}(t) = \mathbf{x}_i(t + s_{k,i})$ are the shifted curves. Necessary conditions for $(\hat{\mathbf{P}}, \hat{\mathbf{S}}, \hat{\mathbf{v}}_1, \ldots, \hat{\mathbf{v}}_K)$ to be a (local) minimizer of (1) are that each of $\hat{\mathbf{P}}$, $\hat{\mathbf{S}}$ and $\hat{\mathbf{v}}_1, \ldots, \hat{\mathbf{v}}_K$ minimizes (1) fixing all the other variables. We prove two key results (see Section S1). The first provides an explicit solution for $\hat{\mathbf{P}}$ given shifts and centers. Importantly, this result holds for any distance $d(\cdot, \cdot)$ and does not rely on any regularity assumption on curves or cluster centers.

Proposition 1. *Fix $\hat{\mathbf{S}}$ and $\hat{\mathbf{v}}_1, \ldots, \hat{\mathbf{v}}_K$. Let $R = \left\{ i \in \{1, \ldots, N\} \,|\, d(\tilde{\mathbf{x}}_{i,\hat{s}_{k,i}}, \hat{\mathbf{v}}_k) > 0 \text{ for all } k \right\}$ and suppose that $|R| \geq K$. Then $\hat{\mathbf{P}} = [\hat{p}_{k,i}]$ is a global minimizer of*

$$J_m(\cdot, \hat{\mathbf{S}}, \hat{\mathbf{v}}_1, \ldots, \hat{\mathbf{v}}_K) \;:\; [0,1]^{K \times N} \longrightarrow \mathbb{R}, \tag{2}$$

under the constraints $\sum_{k=1}^{K} p_{k,i} = 1$, $\forall i$ and $\sum_{i=1}^{N} p_{k,i} > 0$, $\forall k$ if and only if

$$\hat{p}_{k,i} = \left[\sum_{l=1}^{K} \left(\frac{d^2\left(\tilde{\mathbf{x}}_{i,\hat{s}_{k,i}}, \hat{\mathbf{v}}_k\right)}{d^2\left(\tilde{\mathbf{x}}_{i,\hat{s}_{l,i}}, \hat{\mathbf{v}}_l\right)} \right)^{\frac{1}{m-1}} \right]^{-1} \qquad k = 1, \ldots, K \tag{3}$$

for all $i \in R$ *and*

$$\hat{p}_{k,i} = \begin{cases} 0, & k \,:\, d(\tilde{\mathbf{x}}_{i,\hat{s}_{k,i}}, \hat{\mathbf{v}}_k) > 0 \\ \in [0,1], & k \,:\, d(\tilde{\mathbf{x}}_{i,\hat{s}_{k,i}}, \hat{\mathbf{v}}_k) = 0 \end{cases} \tag{4}$$

with $\sum_{k=1}^{K} \hat{p}_{k,i} = 1$*, for all* $i \notin R$*.*

If the i-th curve has positive distance from all cluster centers, (3) states that its probability of belonging to cluster k is inversely proportional to the $(m-1)$-th root of its squared distance from the k-th cluster center. Equation (4) tackles the extreme case, very seldom in practice, of a curve with distance 0 from one or more cluster centers; in this case, the probabilities are set to 0 for all clusters from which the curve has a positive distance. If a curve has distance 0 to exactly one cluster, the constraint implies that the corresponding probability is 1. If a curve has distance 0 to more than one cluster, the corresponding probabilities can be arbitrarily chosen as long as the constraint is satisfied.

The second result provides a formula for $\hat{\mathbf{v}}_1, \ldots, \hat{\mathbf{v}}_K$ given shifts and memberships, and depends on the distance employed. This provides an explicit solution for optimal centers given shifts and probabilities for any distance $d_\alpha(\cdot,\cdot)$ defined as

$$d_\alpha^2(\mathbf{x}, \mathbf{v}) = \sum_{\nu=1}^{d} \frac{w_\nu}{d} \Big[\frac{1-\alpha}{c} \int_0^c \left(x^{(\nu)}(t) - v^{(\nu)}(t) \right)^2 \mathrm{d}t + \frac{\alpha}{c} \int_0^c \left(x'^{(\nu)}(t) - v'^{(\nu)}(t) \right)^2 \mathrm{d}t \Big] \tag{5}$$

where $w_\nu > 0$ is the weight of the ν^{th} component of a d-dimensional curve, indicated by $^{(\nu)}$, $'$ indicates the weak derivative (see, e.g., Evans, 1998, pag. 254), $(0, c)$ is the domain of $\mathbf{v}$, and $\alpha \in [0,1]$ is a parameter that defines the relative weight of the curve's levels and derivatives. When $\alpha = 0$, we require $\mathbf{x}_i \in L^2(\mathbb{R}, \mathbb{R}^d)$ and $\mathbf{v}_k \in V_k = L^2((0, c_k), \mathbb{R}^d)$, where L^2 is the space of square-integrable functions. In this case we obtain an L^2-like distance $d_0(\cdot,\cdot)$ that focuses exclusively on the levels. When $\alpha > 0$, we require $\mathbf{x}_i \in H^1(\mathbb{R}, \mathbb{R}^d)$ and $\mathbf{v}_k \in V_k = H^1((0, c_k), \mathbb{R}^d)$, where H^1 is the Sobolev space of square-integrable functions, with square-integrable first order weak derivative (see, e.g., Evans, 1998, pag. 254). The choice of $\alpha = 1$ leads to an L^2-like pseudo-distance $d_1(\cdot,\cdot)$ focusing on curve variations (their slopes or trends). Finally, $\alpha \in (0,1)$ defines a Sobolev-like distance $d_\alpha(\cdot,\cdot)$ that highlights more complex features of curve shapes, taking into account both levels and variations. Note that no smoothness assumption is needed for the curves nor the cluster centers.

Proposition 2. *Fix* $\hat{\mathbf{P}}$ *and* $\hat{\mathbf{S}}$. *Consider the distance* $d_\alpha(\cdot,\cdot)$, *with fixed* $\alpha \in [0,1]$. *If* $\alpha = 0$, *assume* $\mathbf{x}_i \in L^2(\mathbb{R},\mathbb{R}^d)$ *and* $\mathbf{v}_k \in V_k = L^2((0,c_k),\mathbb{R}^d)$. *If* $\alpha > 0$, *assume* $\mathbf{x}_i \in H^1(\mathbb{R},\mathbb{R}^d)$ *and* $\mathbf{v}_k \in V_k = H^1((0,c_k),\mathbb{R}^d)$. *Then* $\hat{\mathbf{v}}_1,\ldots,\hat{\mathbf{v}}_K$ *are the (unique) global minimizers of*

$$J_m(\hat{\mathbf{P}},\hat{\mathbf{S}},\cdot) \;:\; V_1 \times \cdots \times V_K \longrightarrow \mathbb{R} \tag{6}$$

if and only if

$$\hat{\mathbf{v}}_k = \frac{\sum_{i=1}^{N} (\hat{p}_{k,i})^m \tilde{\mathbf{x}}_{i,\hat{s}_{k,i}}}{\sum_{i=1}^{N} (\hat{p}_{k,i})^m} \qquad a.e.\ in\ (0,c_k),\ \forall k. \tag{7}$$

When $\alpha = 1$, $\hat{\mathbf{v}}_k$ *is defined by (7) up to an additive constant.*

Equation (7) defines the k^{th} cluster center as a weighted average of the shifted curves in $(0,c_k)$. Weights are determined by memberships: the contribution of a curve to the computation of $\hat{\mathbf{v}}_k$ is directly proportional to its probability of belonging to cluster k. Note that if a curve does not belong to any cluster, then the membership probabilities are all around $1/K$, hence the curve has a reduced influence on the definition of the cluster centers. In addition, we implemented a cluster cleaning step in order to make this influence negligible (see Subsection S2.1).

2.2 Algorithm

Propositions 1 and 2 suggest to numerically minimize (1) through an iterative procedure that alternates: i. identification of cluster centers with equation (7), ii. curve alignment (warping function selection), and iii. computation of membership probabilities using equations (3)-(4). We propose the following algorithm for probKMA.

Initialization Fix the number of clusters K and the cluster center lengths $c_1,\ldots,c_K$. Consider an initial membership matrix $\mathbf{P}^{(0)}$ such that $\sum_{k=1}^{K} p_{k,i}^{(0)} = 1$, $\forall i$ and $\sum_{i=1}^{N} p_{k,i}^{(0)} > 0$, $\forall k$ (non-degenerate clusters), and an initial shift matrix $\mathbf{S}^{(0)}$;

Iteration Repeat the following three steps for $iter = 1,2,\ldots,$ until convergence:

i. *Identification of cluster centers.* For each k, compute the k^{th} cluster center $\mathbf{v}_k^{(iter)}$ with equation (7), using the shift $s_{k,i}^{(iter-1)}$ and memberships $p_{k,i}^{(iter-1)}$;

ii. *Curve alignment.* For each i and k, align the curve $\mathbf{x}_i$ to the new cluster center $\mathbf{v}_k^{(iter)}$, selecting the shift $s_{k,i}^{(iter)}$ that minimizes their distance $d\left(\tilde{\mathbf{x}}_{i,s}, \mathbf{v}_k^{(iter)}\right)$;

iii. *Computation of membership probabilities.* Compute the membership matrix $\mathbf{P}^{(iter)}$ with equations (3)-(4), using $\mathbf{v}_k^{(iter)}$ and the shifts $s_{k,i}^{(iter)}$.

Stopping criterion At each iteration, evaluate convergence using the Bhattacharyya distance BC between the membership matrices $\mathbf{P}^{(iter)}$ and $\mathbf{P}^{(iter-1)}$. For each k, compute $BC_k = -\log\left(\sum_{i=1}^{N}\sqrt{p_{k,i}^{(iter)}p_{k,i}^{(iter-1)}}\right)$. Compute BC as the maximum, mean, or order q quantile of all BC_k. Repeat steps i-iii until BC reaches a given tolerance.

Remark 1. Steps i and iii are analogous to the steps of a fuzzy K-means algorithm (Bezdek et al., 1984), or of an EM algorithm for mixture models (Dempster et al., 1977). Steps i and ii correspond to the functional K-means with (global) alignment (Sangalli et al., 2010).

Every iteration can be written in a functional form as

$$\left(\mathbf{P}^{(iter)}, \mathbf{S}^{(iter)}, \mathbf{v}_1^{(iter)}, \ldots, \mathbf{v}_K^{(iter)}\right) \in T_m\left(\mathbf{P}^{(iter-1)}, \mathbf{S}^{(iter-1)}, \mathbf{v}_1^{(iter-1)}, \ldots, \mathbf{v}_K^{(iter-1)}\right)$$

where $T_m : Y \longrightarrow Y$ is the point-to-set map defined by i-iii, and Y the subset of $[0,1]^{K\times N} \times \mathbb{R}^{K\times N} \times V_1 \times \ldots \times V_K$ that satisfies $\sum_{k=1}^{K} p_{k,i} = 1$, $\forall i$ and $\sum_{i=1}^{N} p_{k,i} > 0$, $\forall k$. For each initialization, the algorithm generates a sequence of iterations

$$\left\{T_m^{(iter)}\left(\mathbf{P}^{(0)}, \mathbf{S}^{(0)}, \mathbf{v}_1^{(0)}, \ldots, \mathbf{v}_K^{(0)}\right)\right\}_{iter=1,2,\ldots}. \tag{8}$$

Below, we show that J_m is continuous and descends along (8). This is an important result, which mimics the one in Hathaway et al. (1987) for fuzzy K-means (proof in Section S1).

Lemma 3. *The functional $J_m : Y \longrightarrow \mathbb{R}$ is continuous.*

Theorem 4. *Consider $\mathbf{y}^{(iter-1)} \in Y$. Then for every $\mathbf{y}^{(iter)} \in T_m(\mathbf{y}^{(iter-1)})$ we have*

$$J_m\left(\mathbf{y}^{(iter)}\right) \leq J_m\left(\mathbf{y}^{(iter-1)}\right), \tag{9}$$

i.e. J_m *is a descent functional for* T_m*. Moreover,* J_m *descends strictly along the iterations if* $\mathbf{y}^{(iter-1)} \notin \Omega$*, where* $\Omega \subseteq Y$ *is the solution set of* $\hat{\mathbf{y}} = (\hat{\mathbf{P}}, \hat{\mathbf{S}}, \hat{\mathbf{v}}_1, \ldots, \hat{\mathbf{v}}_K) \in Y$ *such that*

$$J_m\left(\hat{\mathbf{P}}, \hat{\mathbf{S}}, \hat{\mathbf{v}}_1, \ldots, \hat{\mathbf{v}}_K\right) \leq J_m\left(\mathbf{P}, \hat{\mathbf{S}}, \hat{\mathbf{v}}_1, \ldots, \hat{\mathbf{v}}_K\right) \quad \begin{array}{l} \forall\, \mathbf{P} \in [0,1]^{K \times N} \\ \sum_{k=1}^{K} p_{k,i} = 1 \\ \sum_{i=1}^{N} p_{k,i} > 0; \end{array} \tag{10}$$

$$J_m\left(\hat{\mathbf{P}}, \hat{\mathbf{S}}, \hat{\mathbf{v}}_1, \ldots, \hat{\mathbf{v}}_K\right) \leq J_m\left(\hat{\mathbf{P}}, \mathbf{S}, \hat{\mathbf{v}}_1, \ldots, \hat{\mathbf{v}}_K\right) \quad \forall\, \mathbf{S} \in \mathbb{R}^{K \times N}; \tag{11}$$

$$J_m\left(\hat{\mathbf{P}}, \hat{\mathbf{S}}, \hat{\mathbf{v}}_1, \ldots, \hat{\mathbf{v}}_K\right) < J_m\left(\hat{\mathbf{P}}, \hat{\mathbf{S}}, \mathbf{v}_1, \ldots, \mathbf{v}_K\right) \quad \begin{array}{l} \forall\, \mathbf{v}_k \in V_k \\ \mathbf{v}_k \neq \hat{\mathbf{v}}_k. \end{array} \tag{12}$$

Remark 2. Although the previous result does not guarantee that every sequence of iterations (8) converges to a minimizer of the functional J_m, it is a necessary condition for convergence and a desirable property for the algorithm.

In the previous theoretical results and algorithm, the lengths $c_1, \ldots, c_K \in [c_{min}, c_{max}]$ of the cluster centers remain fixed. However, we seek to identify local similarities even when the lengths of the matching curve portions are not known *a priori*. This problem has been already tackled by local sequence alignment methods in bioinformatics, whose goal is to find similar stretches of unknown lengths within a collection of nucleotide or amino acid sequences. In this context, one of the most widely used algorithms is BLAST (Altschul et al., 1990). BLAST starts by finding short stretches shared by the sequences, and uses them as *seeds*. It then extends the seeds on both sides to construct larger local alignments, stopping when the similarity score drops below a given threshold. Borrowing this logic, we add a *center elongation* step to our algorithm. This step is performed only when the algorithm is reaching convergence, to guarantee that we do not extend low-quality cluster centers. We attempt elongation on both the left and the right of each center, generating the elongated center using equation (7) on the corresponding aligned curves, first for the interval $(-\delta_{elong}, c_k)$ and then for the interval $(0, c_k + \delta_{elong})$. For elongation to be acceptable, we require that the corresponding objective function $J_{m,k}(\mathbf{P}, \mathbf{S}, \mathbf{v}_1, \ldots, \mathbf{v}_K) =$

$\sum_{i=1}^{N} (p_{k,i})^m \, d^2\left(\tilde{\mathbf{x}}_{i,s_{k,i}}, \mathbf{v}_k\right)$ decreases or that it increases less than a given threshold $\Delta_{J_{m,k}}$. Note that this implies that Theorem 4 is not valid when the elongation is performed since the objective function J_m is allowed to increase.

Further details on probKMA implementation are provided in Section S2.

3 Cluster evaluation and functional motif discovery

To evaluate a probKMA clustering, we develop a generalized silhouette index, similar to the one used in classic clustering (Rousseeuw, 1987). Our index is defined for portions of curves and measures how well each portion fits its own cluster. First, we dichotomize the membership matrix $\hat{\mathbf{P}}$ to transform it into a matrix of zeros and ones as explained in Subsection S2.1, and we extract all the curve portions belonging to a cluster, i.e. for which the dichotomized membership probability is equal to 1. Next, we compute the distance $d_j(k)$ of each portion $j = 1, \ldots, J$ from cluster k as the mean of the distances between j itself and all the portions of cluster k. We define the intra-cluster distance as $a_j = d_j(k_j)$ – the distance of portion j from the cluster k_j it belongs to – and the inter-cluster distance as $b_j = \min_{k \neq k_j} d_j(k)$ – the minimum distance of portion j from all the other clusters. The generalized silhouette index for portion j is

$$s_j = \frac{b_j - a_j}{\max(b_j, a_j)} \in [-1, 1].$$

Large values of s_j indicate that j is appropriately assigned to its cluster, while low values indicate bad assignments. In particular, negative values signify that portion j is closer to a cluster different from the one it was assigned to. For each cluster k, we then compute its average silhouette index S_k considering all the portions assigned to k. This measures the compactness of the cluster and hence its quality. Finally, the overall average silhouette index S measures the overall quality of the clustering. Similar to classic clustering, silhouettes for portions, clusters and overall clustering are visualized in a silhouette plot (see examples in Figs. S27(b) and S35(c)) that facilitates their interpretation.

Like other K-means algorithms, probKMA finds a local minimum of the functional J_m and its output heavily depends on initialization. If the goal is to locally cluster the

curves in K groups, we repeat the algorithm using different initializations (and possibly different initial lengths) and we select the solution with the lowest value of J_m. When K is not known, the generalized silhouette index allows one to compare the results obtained with different K and select the best one. If the goal is functional motif discovery, we run probKMA multiple times with different initializations, cluster numbers, and motif initial lengths, and form the set of candidate motifs taking the union of the solutions. We clean this set of candidate motifs using generalized silhouette indices and number of occurrences. We then merge very similar candidate motifs, as they may correspond to the same motif identified by multiple runs of probKMA (Section S3). Finally, we utilize a motif search algorithm to locate all instances of the discovered motifs in the input curves, i.e. all portions of curve with distance lower than a given radius from each motif (Section S3).

4 Simulations

4.1 Generating 1-dimensional curves in complex scenarios

Generating curves comprising functional motifs is a non-trivial task since we require motifs to be smoothly embedded in curves while allowing them to occur with noise. To do this, we exploit the flexibility provided by B-splines. We consider a B-spline basis $\{\Phi_l\}_{l=1}^{L}$ of order n, with equally spaced knots $t_1, \dots, t_{L-n+2}$, and define each 1-dimensional curve as $x(t) = \sum_{l=1}^{L} c_l \Phi_l(t)$, where $c_l \in \mathbb{R}$, $l = 1, \dots, L$ are coefficients to be chosen. The order n controls smoothness and complexity of x (x is a curve of class C^{n-2} and a piece-wise polynomial of degree $n-1$). Higher orders provide more degrees of freedom, allowing one to generate curves with more complex shapes, and smoother at the knots. Each Φ_l has compact support – it is 0 outside an interval of length nT, where T is the distance between two subsequent knots; this allows us to define a functional motif of length T fixing the values of n coefficients $c_{\mathrm{m},i}, \dots, c_{\mathrm{m},i+n-1}$ and repeating them multiple times within the same curve or across different curves. Longer motifs of length $2T, 3T, \dots$, that may result in more complex shapes, can be created similarly, fixing the values of $n+1, n+2, \dots$ subsequent coefficients. Since a single curve can embed more than one functional motif, as

well as more than one occurrence of the same motif, we require motifs to be separated by at least one sub-interval (t_i, t_{i+1}) as not to be artificially merged (i.e. we require at least n background coefficients between them). Motif occurrences that are "the same", both in shape and level, are generated adding Gaussian noise to their coefficients: $\tilde{c}_{\mathrm{m},j} = c_{\mathrm{m},j} + \epsilon_j$, $\epsilon_j \overset{iid}{\sim} N(0, \sigma^2)$. Motif occurrences that are "the same" in shape but have different levels are obtained adding a constant δ_{m} to all the coefficients that define a single occurrence (different constants for different occurrences): $\tilde{c}_{\mathrm{m},j} = c_{\mathrm{m},j} + \delta_{\mathrm{m}} + \epsilon_j$, $\epsilon_j \overset{iid}{\sim} N(0, \sigma^2)$. Background coefficients $c_{\mathrm{bg}} \in [a, b]$ (i.e., coefficients not corresponding to motifs) are generated as $(c_{\mathrm{bg}} - a)/b \overset{iid}{\sim} Beta(0.45, 0.45)$, creating reasonably different backgrounds for both the curve and its derivative. With this flexible model, we can generate data in several scenarios, varying curve and motif lengths, as well as variability, frequencies, and positions of motifs.

4.2 Functional motif discovery: varying curve length ℓ and noise σ in motifs

This simulation study aims to demonstrate the performance of probKMA in discovering functional motifs embedded in a set of curves and to examine the effects of increasing curve length and the noise level comprised in motif occurrences. We consider two different scenarios, with sets of curves embedding (1) motifs that share both shapes and levels; or (2) motifs that share shapes but have different levels.

In scenario (1), we consider a set of 20 curves embedding two functional motifs, each with 12 occurrences (see Fig. 2 and Figs. S3-S5). In particular, 12 curves contain only one occurrence of a motif (6 curves for each of the two motifs), 4 curves contain two occurrences of a motif (2 curves for each of the two motifs), 2 curves contain one occurrence of each of the two motifs, and 2 curves contain no motif occurrences at all. We generate data using a B-spline basis of order 3, knots at distance 10, and motifs of length 60. Coefficients defining the two motifs are randomly generated from a $Beta(0.45, 0.45)$ distribution rescaled to $[-15, 15]$. We consider four different curve lengths $\ell = 200, 300, 400, 500$ and four levels of noise $\sigma = 0.1, 0.5, 1, 2$, for a total of 16 simulated datasets. In order to maximize the consistency among these datasets and thus highlight the effects of different ℓ and σ values,

we place motif occurrences within the leftmost sub-interval of length 200 of each curve that is common to all datasets, utilizing the same motif positions and background in all 16 cases. We treat the simulated curves as known, and we sample them on a grid of points at distance 1, so that each motif corresponds to 61 points. For each combination of ℓ and σ, we run our probKMA-based functional motif discovery with Sobolev-like distance $d_{0.5}(\cdot, \cdot)$. We evaluate the number of motifs found, the distance between true and estimated motifs, the estimated lengths of motifs, and the number of true and false positives. ProbKMA is run for $K = 2, 3$, minimum motif lengths $c_{min} = 40, 50, 60$ and 20 random initializations for each (K, c_{min}) pair (the maximum motif length is set to 70; see Subsection S4.1 for other parameters). The same initializations are employed for all ℓ and σ combinations. Results for $\ell = 200$ can be found in Fig. 3 and show very good performance for our method. As expected, performance slightly declines when more noise is introduced in the motif instances: some occurrences can be missed, and/or false positives can be included. However, results remain satisfactory even when $\sigma = 2$. Results for other curve lengths

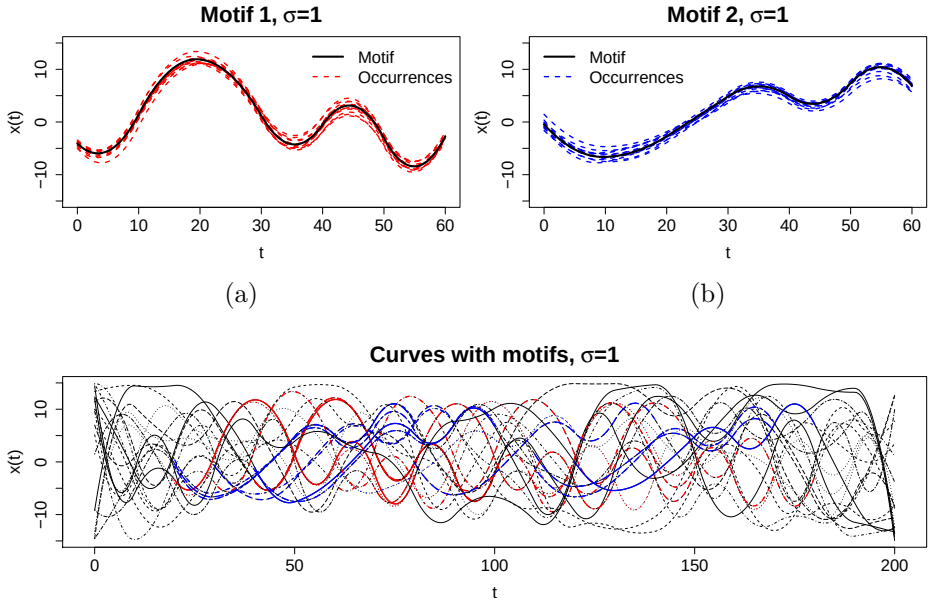
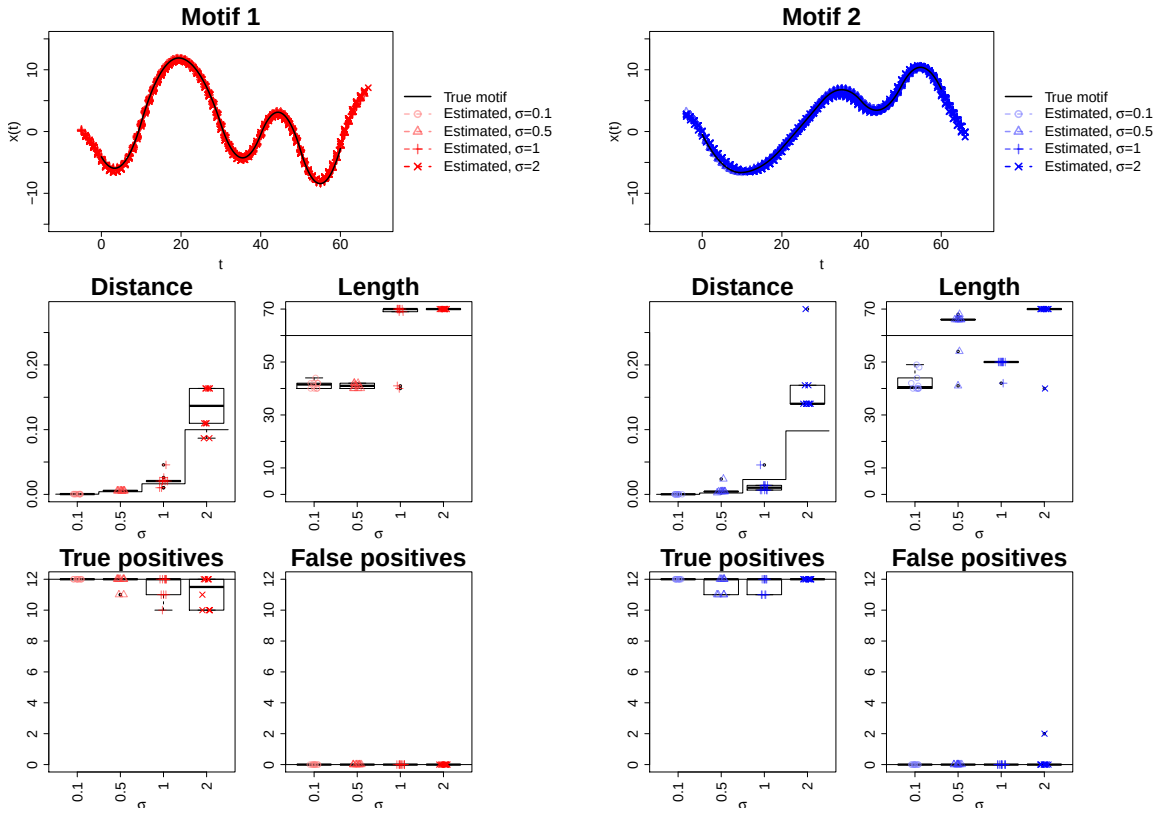


Figure 2: Simulation scenario (1) with $\ell = 200$ and $\sigma = 1$. (a), (b) Two functional motifs (solid curves) and 12 aligned occurrences of each (dashed curves); (c) 20 curves embedding occurrences of the two motifs.

are shown in Figs. S6-S8. They suggest the same behavior as the noise level increases and they appear rather robust across lengths. The only effect of increasing the ratio between background curve portions and curve portions occupied by motifs is a slight increase in false positives, which occurs exclusively when also the noise level is high.

In scenario (2) we consider the same curves and motifs as in scenario (1), but allow motif occurrences to have different levels (see Figs. S9-S12). In particular, a random value $\delta_m \sim U(-10, 10)$ is added to all the coefficients defining each motif occurrence (a different value δ_m for each occurrence). For each combination of ℓ and σ, we run our probKMA-based discovery with the L^2-like pseudo-distance $d_1(\cdot, \cdot)$ to focus on curve variation. Parameters



(a) (b)

Figure 3: Functional motif discovery results for simulation scenario (1) with $\ell = 200$ and varying σ. (a) Motif 1; (b) Motif 2. The boxplots are obtained from 10 replications at each σ value. They show the distance between true and estimated motifs (stepwise line: distance between the true motif and the average of all motif occurrences), the estimated length of motifs, true and false positives. In all cases, exactly 2 motifs are found.

are the same as in scenario (1), and detailed results are provided in Figs. S13-S16. We find again that our method has good performance, affected (as expected) by the noise level, but not much by the length of the curves. In some cases, especially when the curves are very long, we actually discover motifs that were not embedded in the simulated data. Note that, strictly speaking, these motifs are not altogether false. As one elongates the background portions of the curves, it is possible to generate by chance a few patterns that recur often enough to be identified by our algorithm. In our experiments, these additional motifs are noisier and have fewer occurrences than the two motifs originally embedded in the data.

Additional simulation studies to validate the results described above and examine the robustness of the method to the number of initializations can be found in Subsection S4.1, in particular in Figs. S17-S22.

4.3 Comparison with time series motif discovery

We compare our probKMA-based functional motif discovery to time series motif discovery. In particular, we consider the recent Matrix Profile (Yeh et al., 2016, 2018). This tool discovers the top motif pairs in a time series and, for each of these pairs, provides all the neighboring subseries, i.e. all the subseries with distance less than R from the motif pair (see Subsection S4.2). We consider two specifications of the simulation scenarios introduced in Subsection 4.2: the simple case of short curves and low noise level ($\ell = 200$ and $\sigma = 0.1$), and the complex case of long curves and high noise level ($\ell = 500$ and $\sigma = 2$). Both probKMA-based motif discovery and Matrix Profile discover the two motifs in the simple case. However, when curves are longer and motifs noisier (complex case), Matrix Profile fails to find Motif 1 and includes many false positives in Motif 2. When the radius is large, it does correctly identify a small number of occurrences of Motif 1, but it also reports a very large number of false positives (for both motifs). On the contrary, probKMA-based motif discovery remains robust to noise level and curve length, and is able to identify both motifs with a very small number of false positives (see Tables S1-S3).

4.4 Comparison with non-sparse and sparse functional clustering

We perform simulation experiments to compare probKMA, meant as a clustering method and separate from its motif discovery purpose, to other functional clustering methods: the standard functional K-means (Tarpey and Kinateder, 2003), the K-means with (global) alignment of Sangalli et al. (2010), and the sparse clustering technique of Floriello and Vitelli (2017) (see details in Subsection S4.3). We consider the following scenarios: (a) curves in the two clusters are aligned and they differ on the entire domain; (b) curves in the two clusters are misaligned and they differ on the entire domain; (c) curves in the two clusters differ on a portion of the domain and this portion is aligned; (d) curves in the two clusters differ on a portion of the domain and this portion is misaligned. Running all methods with Euclidean distance and $K = 2$, they all correctly classify curves in scenario (a). K-means only works in this scenario, while K-means with (global) alignment performs well in scenarios (a) and (b), and sparse clustering performs well in scenarios (a) and (c). Interestingly, when the noise level is small, sparse clustering also achieves a good performance in scenario (b). ProbKMA performs very well in all scenarios (Tables S4-S5).

5 Real data applications

ProbKMA is very flexible and it can be applied to any kind of functional data, from any domain. As shown in the previous section, it can be employed not only to discover functional motifs but also as a (probabilistic) local clustering method. In this section, we provide two detailed real data applications which illustrate these two possible uses of probKMA. An additional application to a well-known dataset in the functional clustering literature, the Berkeley Growth Study curves, is provided in Subsection S5.1.

5.1 Local clustering of Italian Covid-19 excess mortality curves

Italy was the first European country to be hit by the Covid-19 pandemic, with the first confirmed cases around mid-February 2020. Italian regions were hit at different times and with different strength, and local authorities implemented different responses, especially in

the initial stages. Comparing the pandemic evolution across regions can therefore provide important insights on the role of underlying factors and different containment measures. We estimate excess mortality due to Covid-19 in Italy using the mortality data (due to all causes) from the Italian Institute of Statistics (ISTAT). The dataset contains the daily number of deaths for 7,270 municipalities (covering about 93.5% of the Italian population) from January 1^{st} to April 30^{th}, for the years 2015-2020. We aggregate data by region and we compute the excess mortality rate curves as the daily difference between 2020 deaths and average deaths in the period 2015-2019, divided by the population of the considered municipalities (see Subsection S5.2). In order to focus on the Covid-19 period, we only consider data starting from February 16^{th}. To reduce noise, we smooth the curves using B-spline smoothing (cubic splines, knots at each day, roughness penalty on the second derivative, and smoothing parameter chosen by average generalized cross-validation). Smoothed curves are shown in Fig. 4(a), while raw data are in Fig. S29.

We cluster the 20 Italian regions according to their excess mortality rate curves to assess if some regions are sharing similar patterns (see also Boschi et al., 2021). We are interested in the entirety of the curves – possibly excluding the extremes of their domains – but we allow shifts in their alignment to take into consideration possible differences in the time when the (shared) patterns began in each region. We employ probKMA as a local clustering method, with L^2-like distance $d_0(\cdot, \cdot)$ and cluster centers of fixed length $c = 65$ days (hence allowing for a maximum shift of 10 days). Fig. 4(b)-(c) shows probKMA results for $K = 2$, when assigning each curve to the cluster with highest membership probability. Cluster 2 contains the regions (mainly located in the north of Italy) where Covid-19 hit the hardest. Lombardia is the region with earliest Covid-19 related deaths, followed by Emilia Romagna, Marche, Liguria, Piemonte and Trento/Bolzano, and last Valle d'Aosta (with a delay of 7 days). Cluster 1 contains the regions with milder epidemic patterns. Interestingly, Veneto is placed in Cluster 1 despite being the first region, together with Lombardia, to report Covid-19 cases. This suggests that Veneto successfully managed to flatten the curve with its early mass testing and contact tracing response (Mugnai and Bilato, 2020). In contrast, the pattern in Lombardia is so stark that it does not seem to fit

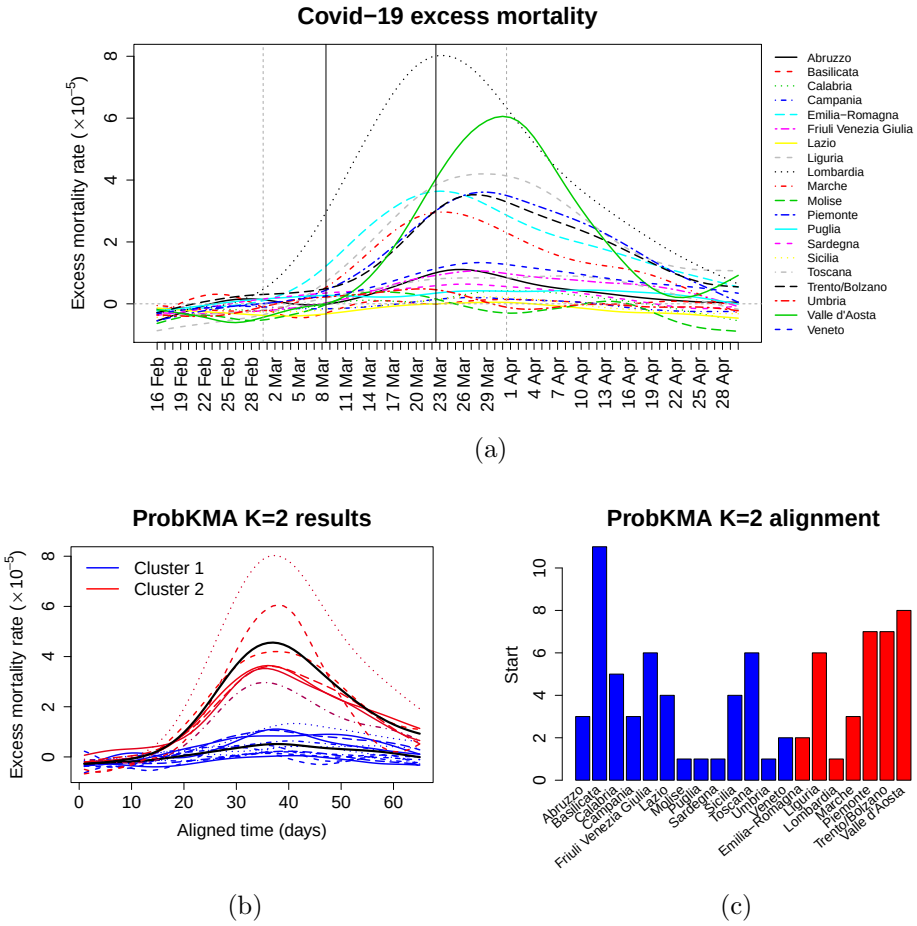


Figure 4: Covid-19 excess mortality curves and probKMA results. (a) Smoothed curves. Vertical solid lines represent national lockdown (March 9^{th}) and closure of all non-essential economic activities (March 23^{rd}); (b) Cluster centers (thick curves) with aligned portions of curves; (c) Alignment between portions of curves within clusters (start day of each portion).

properly even in Cluster 1 and shows a large distance from the cluster center (Fig. S30). Indeed, repeating the clustering with $K = 3$, Lombardia is placed in a cluster of its own, while the other two clusters and the alignments within them do not change (see Fig. S31).

5.2 Motif discovery in mutagenesis data

To fully illustrate the proposed method in its motif discovery purpose, we apply it to a mutagenesis dataset adapted from Kuruppumullage Don et al. (2013) that we provide at `https://github.com/marziacremona/mutagenesis_data`. Mutagenesis comprises all the

processes by which mutations are generated in DNA, it is one of the major evolutionary forces and is central to causing many human diseases (e.g. cancer). Understanding mutagenesis and how it is influenced by the genomic landscape is key to shedding light on genome dynamics (Makova and Hardison, 2015). Kuruppumullage Don et al. (2013) estimated different types of neutral (i.e. not affected by selection) mutation rates in non-overlapping windows along the human genome comparing it with primates, and employed Hidden Markov Models to define six divergence states and segment the genome accordingly. One of the states is of particular interest: it comprises hot regions with very high rates for substitutions, small insertions, and small deletions, which are associated with high GC (guanine-cytosine) content, early replication timing, and open chromatin. Since these results were obtained at a rather large scale (1-Mb windows), investigating rates at a finer resolution within the hot regions may reveal more specific trends and patterns of variation. Note that, in general, we could aim at discovering 3-dimensional motifs, i.e. joint patterns of substitutions, insertions, and deletions. However, for simplicity, we consider only 1-dimensional substitution rate curves. Estimating high-resolution substitution rates in 1-kb windows within each hot region (with the same pipeline as in Kuruppumullage Don et al., 2013, see Subsection S5.3 and Fig. S32), we generate a dataset of 43 curves, varying in length from 1 Mb (corresponding to a grid of $1,000$ points) to 22 Mb ($22,000$ points). The curves are very noisy and contain several missing or inaccurate values since in many 1-kb windows the information needed to estimate rates is scarce (see Fig. S33). In particular, substitution rates can be reliably estimated only in 60% of the 1-kb windows. After pre-processing with stochastic regression imputation and local smoothing, missing values are reduced to 17% of the windows (see Subsection S5.3).

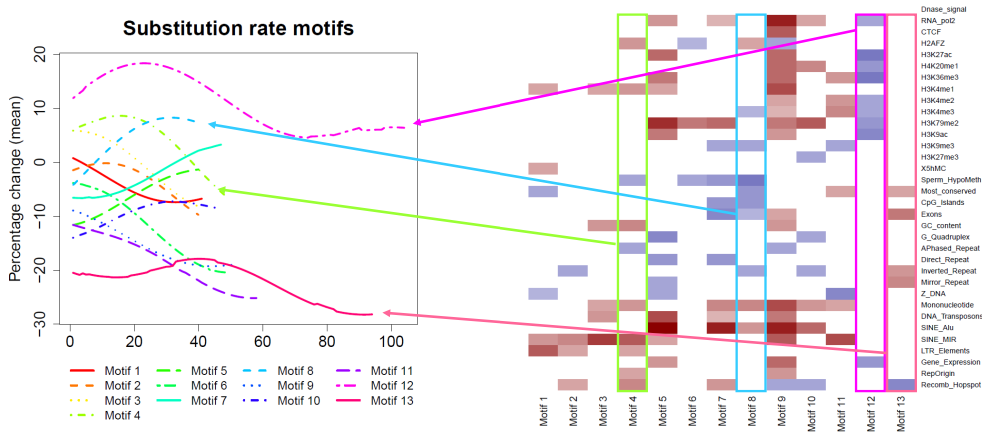
We employ our probKMA-based functional motif discovery on the 43 curves using the Sobolev-like distance $\tilde{d}_{0.5}(\cdot,\cdot)$ (the generalized version which can accommodate large gaps; see Subsection S2.2). We look for motifs with minimum lengths $c_{min} = 40, 50, 60, 70$ (maximum length $c_{max} = 150$), and we run probKMA for $K = 2, 3, 4, 5$ using 10 random initialization for each (K, c_{min}) pair. We employ our generalized silhouette index to evaluate each probKMA run and to filter the set of candidate motifs, and we select motif-specific

Table 1: ProbKMA-based functional motif discovery in substitution rate curves. For each motif found, we report the number of occurrences and their mean distance from the motif.

Motif	1	2	3	4	5	6	7	8	9	10	11	12	13
Number	19	12	27	37	63	12	72	47	14	11	9	8	6
Mean dist	1.9	1.9	3.5	5.1	6.5	3.0	8.7	7.4	5.2	3.0	5.5	18.5	17.0

radii based on probKMA results (see details in Subsection S5.3). We identify 13 functional motifs that differ substantially in length (40 to 104 kb), levels and shapes (see Fig. 5(a)). The motifs also differ in frequency (i.e. number of occurrences in the data) and level of variability (see Table 1). This highlights the advantage of employing a motif discovery methodology able to learn motif-specific length, frequency, and variability from the data.

At least four of the motifs found are of biological interest: Motif 12 corresponds to eight long sub-regions (about 100 kb) with extremely high substitution rate (an elevation of 10% to 20% relative to the mean level across all hot regions, which is already elevated in comparison to the genome at large). Motif 4 and Motif 8 also present very high substitution rate and opposite patterns. In Motif 4, rate is about 10% above the overall hot regions mean for the initial $\sim$20 kb, and then decrease. In Motif 8 rate increases and then stabilizes at about 10% above the mean for $\sim$20 kb. The two motifs have similar variability and are both very frequent (37 and 47 occurrences, respectively). Finally, Motif 13 corresponds to six long sub-regions with a substitution rate 20-30% below the mean. These portions of the hot regions are in fact not hot; substitutions rate is similar to that of the rest of the genome. To investigate the genomic landscape of the motifs found, we consider a set of 35 genomic features measured in each of the 1-kb windows constituting the hot regions. These features represent biological contexts that have an interplay with mutagenesis, such as DNA conformation, DNA sequence, replication, recombination, chromatin openness and modifications (see Table S6). We then compare, independently for each genomic feature and each motif, the mean of the measurements in motif occurrences with the mean across all hot regions. We perform a simulation-based two-sided test for mean difference, where the empirical null distribution is obtained from 1000 datasets generated by randomly relocating motif occurrences within the set of curves. Fig. 5(b) shows that each motif has a



(a) (b)

Figure 5: ProbKMA-based functional motif discovery in substitution rate curves. (a) Motifs found, plotted as percent changes with respect to the mean substitution rate across all hot regions; (b) Genomic landscape of the motifs, with color intensity proportional to the significance ($-\log_{10}(p)$) of a mean difference two-sided test contrasting motif occurrences and hot regions at large; red, blue, and white represent positive, negative and non-significant ($p > 0.1$) differences, respectively. Rimmed columns and arrows show four particularly interesting motifs.

characteristic genomic landscape, which helps in its biological interpretation. For example, occurrences of Motif 13 are enriched in exons and conserved elements compared to hot regions in general; their lowered substitution rate may correlate with such enrichments.

6 Discussion

This article, for the first time to the best of our knowledge, tackles the problem of *functional motif discovery* from a statistical perspective. We proposed probKMA for discovering candidate motifs in a set of curves, incorporating ideas from functional data analysis, bioinformatics and fuzzy clustering. In addition, we proposed a generalized silhouette index to evaluate probKMA results, and implemented a post-processing for merging candidate motifs and searching motif occurrences along the curves. Although many alternative strategies can be employed in post-processing, each with pros and cons, results on simulated and real

data suggest that our implementation is effective in a range of scenarios.

ProbKMA employs a flexible definition of curve similarity, which incorporates both levels and derivatives. In addition, similarity is defined locally, in a way that tolerates large gaps in the curves. This broadens the application scope of our methodology. ProbKMA can also be applied to multivariate curves and does not require the user to specify the exact motif lengths or the motif variability levels at the outset. These are learned from the data – the user only needs to specify the minimum and the maximum lengths of the motifs to discover – substantially improving performance with respect to approaches where lengths and/or radii are fixed. Real data applications usually require some pre-processing steps – such as smoothing to estimate the curves from discretely observed data – which might artificially introduce "false" motifs in the curves (see, e.g., Subsection S5.3). As a consequence, the user must carefully select minimum motif lengths which are compatible with the pre-processing, in particular with the choice of smoothing parameters. The minimum motif length ought to be larger than the length of potential artificially-introduced motifs which, intuitively, is larger the more smoothing has been applied to the data.

In our experience, motif discovery with probKMA can fail when motifs are too similar to one another or when they are too similar to background portions of the curves. This can happen by chance when motifs are very noisy. Relatedly, simulations show that, when motifs are very noisy and/or dispersed in very long curves, our method can identify motifs that were not intentionally introduced in the data, but rather randomly created when generating background portions of the curves. In a way, these additional motifs may be considered as unintentional and yet true (as opposed to false) positives; they do recur in the curves in a way that is detectable by the algorithm. Nevertheless, in our simulations they are noisier and have fewer occurrences. This observation underscores the need for further work addressing the statistical significance of the motifs. The flexible model that we introduced to generate simulation data may play an important role in this context, providing a way to estimate the likelihood of discovering motifs in background curves.

We used a deliberately general, data-driven and non-parametric notion of functional motif – in line with those used in, e.g., the bioinformatics and data mining literature

(Bailey et al., 2006; Lin et al., 2002). Albeit beyond the scope of this article, it would be of utmost interest to formulate a parametric definition of functional motif and develop a rigorous statistical theory for its estimation.

Separately from its motif discovery purpose, probKMA can also be employed for probabilistic clustering of misaligned functional data based on local similarities. In this respect, it also represents a generalization of sparse clustering procedures recently proposed in functional data analysis (Fraiman et al., 2016; Floriello and Vitelli, 2017). In the limit, when the minimum motif length is close to the length of the curves under consideration, probKMA becomes a probabilistic version of K-means with (global) alignment (Sangalli et al., 2010).

Supplementary material

Supplementary material includes proofs, additional methods and results. An R implementation (with examples) is available at `https://github.com/marziacremona/ProbKMA-FMD`.

References

Altschul, S. F., W. Gish, W. Miller, E. W. Myers, and D. J. Lipman (1990). Basic local alignment search tool. *Journal of Molecular Biology 215*(3), 403–410.

Bailey, T. L., N. Williams, C. Misleh, and W. W. Li (2006). MEME: discovering and analyzing dna and protein sequence motifs. *Nucleic Acids Research 34*(suppl_2), W369–W373.

Bezdek, J. C. (1981). *Pattern recognition with fuzzy objective function algorithms*. Springer.

Bezdek, J. C., R. Ehrlich, and W. Full (1984). FCM: the fuzzy c-means clustering algorithm. *Computers & Geosciences 10*(2), 191–203.

Boschi, T., J. Di Iorio, L. Testa, M. A. Cremona, and F. Chiaromonte (2021). Functional data analysis characterizes the shapes of the first COVID-19 epidemic wave in Italy. *Scientific Reports 11*, 17054.

Cremona, M. A., L. M. Sangalli, S. Vantini, G. I. Dellino, P. G. Pelicci, P. Secchi, and L. Riva (2015, 10). Peak shape clustering reveals biological insights. *BMC Bioinformatics 16*(1), 349.

Dempster, A. P., N. M. Laird, and D. B. Rubin (1977). Maximum likelihood from incomplete data via the EM algorithm. *Journal of the Royal Statistical Society. Series B 39*(1), 1–38.

Evans, L. C. (1998). *Partial differential equations*.

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

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

Fraiman, R., Y. Gimenez, and M. Svarc (2016). Feature selection for functional data. *Journal of Multivariate Analysis 146*, 191–208.

Friedman, J. H. and J. J. Meulman (2004). Clustering objects on subsets of attributes. *Journal of the Royal Statistical Society. Series B 66*(4), 815–849.

Hathaway, R., J. Bezdek, and W. Tucker (1987). An improved convergence theory for the fuzzy isodata clustering algorithms. *Analysis of fuzzy information 3*, 123–132.

Horváth, L. and P. Kokoszka (2012). *Inference for functional data with applications*, Volume 200. Springer Science & Business Media.

Jacques, J. and C. Preda (2014). Functional data clustering: a survey. *Advances in Data Analysis and Classification 8*(3), 231–255.

Kuruppumullage Don, P., G. Ananda, F. Chiaromonte, and K. D. Makova (2013). Segmenting the human genome based on states of neutral genetic divergence. *Proceedings of the National Academy of Sciences 110*(36), 14699–14704.

Lin, J., E. Keogh, S. Lonardi, and P. Patel (2002). Finding motifs in time series. In *8th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, Edmonton, Alberta, Canada*.

Linardi, M., Y. Zhu, T. Palpanas, and E. Keogh (2018). Matrix profile X: VALMOD - scalable discovery of variable-length motifs in data series. In *ACM SIGMOD/PODS International Conference on Management of Data / Principles of Database Systems, Houston, Texas, USA*.

Liu, X. and M. C. Yang (2009). Simultaneous curve registration and clustering for functional data. *Computational Statistics & Data Analysis 53*(4), 1361–1376.

Makova, K. D. and R. C. Hardison (2015). The effects of chromatin organization on variation in mutation rates in the genome. *Nature Reviews Genetics 16*(4), 213.

Mueen, A., E. Keogh, Q. Zhu, S. Cash, and B. Westover (2009). Exact discovery of time series motifs. In *SIAM International Conference on Data Mining, Sparks, Nevada, USA*.

Mugnai, G. and C. Bilato (2020). Covid-19 in italy: lesson from the Veneto region. *European Journal of Internal Medicine*.

Park, J. and J. Ahn (2017). Clustering multivariate functional data with phase variation. *Biometrics 73*(1), 324–333.

Ramsay, J. O. and B. W. Silverman (2005). *Functional data analysis* (2 ed.). Springer.

Rousseeuw, P. J. (1987). Silhouettes: a graphical aid to the interpretation and validation of cluster analysis. *Journal of Computational and Applied Mathematics 20*, 53–65.

Sangalli, L. M., P. Secchi, S. Vantini, and V. Vitelli (2010). K-mean alignment for curve clustering. *Computational Statistics & Data Analysis 54*(5), 1219–1233.

Tarpey, T. and K. K. Kinateder (2003). Clustering functional data. *Journal of Classification 20*(1), 093–114.

Vitelli, V. (2019). A novel framework for joint sparse clustering and alignment of functional data. *arXiv*, 1912.00687.

Witten, D. M. and R. Tibshirani (2010). A framework for feature selection in clustering. *Journal of the American Statistical Association 105*(490), 713–726.

Yeh, C.-C. M., Y. Zhu, L. Ulanova, N. Begum, Y. Ding, H. A. Dau, Z. Zimmerman, D. F. Silva, A. Mueen, and E. Keogh (2018). Time series joins, motifs, discords and shapelets: a unifying view that exploits the matrix profile. *Data Mining and Knowledge Discovery 32*(1), 83–123.

Yeh, C. M., Y. Zhu, L. Ulanova, N. Begum, Y. Ding, H. A. Dau, D. F. Silva, A. Mueen, and E. Keogh (2016). Matrix profile I: all pairs similarity joins for time series: A unifying view that includes motifs, discords and shapelets. In *IEEE 16th International Conference on Data Mining, Barcelona, Spain*.

Probabilistic K-means with local alignment for clustering and motif discovery in functional data

Supplementary material

Marzia A. Cremona *
Dept. of Operations and Decision Systems, Université Laval
and
Francesca Chiaromonte
Dept. of Statistics, The Pennsylvania State University

November 30, 2022

Abstract

Supplementary material includes proofs and additional details on implementation, post-processing, simulations, and real-data analyses.

*M.A. Cremona is also affiliated to CHU de Québec – Université Laval Research Center and Dept. of Statistics, Penn State University. F. Chiaromonte is also affiliated to the Inst. of Economics and EMbeDS, Sant'Anna School of Advanced Studies.

S1 Proofs

Proof of Proposition 1. We start considering the minimization of the functional (2) ignoring the constraint $\sum_{i=1}^{N} p_{k,i} > 0$ for all k. The membership probabilities $\mathbf{p}_i = [p_{1,i}, \ldots, p_{K,i}]$ related to different curves $\mathbf{x}_i$ vary independently and

$$\min_{\substack{p_{k,i}\in[0,1] \\ \sum_{k=1}^{K} p_{k,i}=1}} J_m(\mathbf{P}, \hat{\mathbf{S}}, \hat{\mathbf{v}}_1, \ldots, \hat{\mathbf{v}}_K) \;=\; \sum_{i=1}^{N} \min_{\substack{p_{k,i}\geq 0 \\ \sum_{k=1}^{K} p_{k,i}=1}} \sum_{k=1}^{K} (p_{k,i})^m d^2(\tilde{\mathbf{x}}_{i,\hat{s}_{k,i}}, \hat{\mathbf{v}}_k).$$

Hence the optimization problem is equivalent to N independent minimizations of the functions $f_{m,i} : \mathbb{R}^K \longrightarrow \mathbb{R}$, $\mathbf{p}_i \longmapsto \sum_{k=1}^{K} (p_{k,i})^m d^2(\tilde{\mathbf{x}}_{i,\hat{s}_{k,i}}, \hat{\mathbf{v}}_k)$ subject to the constraints $p_{k,i} \geq 0$ and $\sum_{k=1}^{K} p_{k,i} = 1$, for any $i = 1, \ldots, N$.

For $i \notin R$, $f_{m,i}$ with constraints $p_{k,i} \geq 0$ and $\sum_{k=1}^{K} p_{k,i} = 1$ is minimized by $\hat{\mathbf{p}}_i$ if and only if it satisfies (4). In particular, $f_{m,i}(\hat{\mathbf{p}}_i) = 0$.

For $i \in R$, we employ Karush-Kuhn-Tucker conditions on $f_{m,i}$ with $g_k(\mathbf{p}_i) = -p_{k,i} \leq 0$ for all k and $h(\mathbf{p}_i) = \sum_{k=1}^{K} p_{k,i} - 1 = 0$. Regularity conditions are satisfied ($f_{m,i}$, g_k and h are continuously differentiable, g_k and h are affine functions) and the Lagrangian associated to the optimization problem is given by $\mathcal{L}(\mathbf{p}_i, \lambda, \mu) = f_{m,i}(\mathbf{p}_i) - \sum_{k=1}^{K} \lambda_k p_{k,i} + \mu \left(\sum_{k=1}^{K} p_{k,i} - 1 \right)$. If $\hat{\mathbf{p}}_i$ is a constrained minimizer of $f_{m,i}$ then there exist constants $\hat{\lambda}$ and $\hat{\mu}$, with $(\hat{\lambda}, \hat{\mu}) \neq \mathbf{0}$, such that the following conditions hold:

$$\nabla f_{m,i}(\mathbf{p}_i) - \sum_{k=1}^{K} \hat{\lambda}_k \nabla p_{k,i} + \hat{\mu} \nabla \left(\sum_{k=1}^{K} p_{k,i} - 1 \right) \Bigg|_{\mathbf{p}_i = \hat{\mathbf{p}}_i} = \mathbf{0}, \tag{S1}$$

$$\hat{\lambda}_k \hat{p}_{k,i} = 0 \qquad k = 1, \ldots, K, \tag{S2}$$

$$\hat{\lambda}_k \geq 0 \qquad k = 1, \ldots, K. \tag{S3}$$

Condition (S1) implies $\frac{\partial f_{m,i}}{\partial p_{k,i}}(\hat{\mathbf{p}}_i) - \hat{\lambda}_k + \hat{\mu} = 0$ for each k, and hence

$$\hat{p}_{k,i} = \left(\frac{\hat{\lambda}_k - \hat{\mu}}{m\, d^2\left(\tilde{\mathbf{x}}_{i,\hat{s}_{k,i}}, \hat{\mathbf{v}}_k\right)} \right)^{\frac{1}{m-1}}.$$

From $\hat{p}_{k,i} \geq 0$ follows $\hat{\lambda}_k \geq \hat{\mu}$ for all k. Suppose there exists l such that $\hat{\lambda}_l > 0$. Then condition (S2) implies $\hat{p}_{l,i} = 0$, hence $\hat{\lambda}_l = \hat{\mu}$. For all k, we obtain $0 < \hat{\lambda}_l = \hat{\mu} \leq \hat{\lambda}_k$ and hence, from condition (S2), $\hat{p}_{k,i} = 0$. This solution is not admissible because it does not

respect the constraint $\sum_{k=1}^{K} p_{k,i} = 1$, hence $\hat{\lambda} = \mathbf{0}$, $\hat{\mu} < 0$ and $\hat{p}_{k,i} = \left(\dfrac{-\hat{\mu}}{m\, d^2\left(\tilde{\mathbf{x}}_{i,\hat{s}_{k,i}}, \hat{\mathbf{v}}_k\right)} \right)^{\frac{1}{m-1}}$, $k = 1, \ldots, K$. In order to compute the value of $\hat{\mu}$ we use the constraint $\sum_{k=1}^{K} p_{k,i} = 1$, obtaining

$$\hat{p}_{k,i} = \left[\sum_{l=1}^{K} \left(\frac{d^2\left(\tilde{\mathbf{x}}_{i,\hat{s}_{k,i}}, \hat{\mathbf{v}}_k\right)}{d^2\left(\tilde{\mathbf{x}}_{i,\hat{s}_{l,i}}, \hat{\mathbf{v}}_l\right)} \right)^{\frac{1}{m-1}} \right]^{-1} \qquad k = 1, \ldots, K.$$

To show that the previous equation is also sufficient for $\hat{\mathbf{p}}_i$ to be a constrained minimizer of $f_{m,i}$, we observe that $f_{m,i}$, g_k and h are twice continuously differentiable and we consider the Hessian matrix $H_{\mathcal{L}} = \left[\frac{\partial^2 \mathcal{L}_{m,i}}{\partial p_{l,i} p_{k,i}} \right]$ of the Lagrangian. For $l \neq k$ we have $\frac{\partial^2 \mathcal{L}_{m,i}}{\partial p_{l,i} p_{k,i}} = 0$, while for $l = k$ we have $\frac{\partial^2 \mathcal{L}_{m,i}}{\partial^2 p_{k,i}}(\mathbf{p}_i, \lambda, \mu) = m(m-1)(p_{k,i})^{m-2} d^2\left(\tilde{\mathbf{x}}_{i,\hat{s}_{k,i}}, \hat{\mathbf{v}}_k\right)$. The diagonal matrix $H_{\mathcal{L}}$ is a positive definite matrix in the point $(\hat{\mathbf{p}}_i, \hat{\lambda}, \hat{\mu})$ that satisfies the first-order Karush-Kuhn-Tucker conditions (since $m > 1$ and $\hat{p}_{k,i} > 0$), hence $\hat{\mathbf{p}}_i$ is a strict local minimizer of $f_{m,i}$.

The set $\left\{ \mathbf{p}_i \in \mathbb{R}^K \,|\, p_{k,i} \in [0,1],\ \sum_{k=1}^{K} p_{k,i} = 1 \right\}$ is convex and the function $f_{m,i}$ is strictly convex (since $m > 1$ and $d(\tilde{\mathbf{x}}_{i,\hat{s}_{k,i}}, \hat{\mathbf{v}}_k)) > 0$ for all k), hence the local minimizer $\hat{\mathbf{p}}_i$ is actually the unique global minimizer of $f_{m,i}$.

Finally, the hypothesis $|R| \geq K$ guarantees that the solution $\hat{\mathbf{P}}$ defined by (3)-(4) satisfy the constraint $\sum_{i=1}^{N} \hat{p}_{k,i} > 0$ for all k. □

Proof of Proposition 2. Since the contribution of each cluster k to the functional (6) is independent of the contributions of other clusters, we have

$$\min_{\mathbf{v}_1, \ldots, \mathbf{v}_K} J_m(\hat{\mathbf{P}}, \hat{\mathbf{S}}, \mathbf{v}_1, \ldots, \mathbf{v}_K) \;=\; \sum_{k=1}^{K} \min_{\mathbf{v}_1, \ldots, \mathbf{v}_K} \sum_{i=1}^{N} (\hat{p}_{k,i})^m d_\alpha^2\left(\tilde{\mathbf{x}}_{i,\hat{s}_{k,i}}, \mathbf{v}_k\right).$$

Hence we can solve K independent optimization problems. In particular, for each cluster k we minimize $g_{m,k} : V_k \longrightarrow \mathbb{R}$, $\mathbf{v}_k \longmapsto \sum_{i=1}^{N} (\hat{p}_{k,i})^m d_\alpha^2\left(\tilde{\mathbf{x}}_{i,\hat{s}_{k,i}}, \mathbf{v}_k\right)$. Let $\hat{\mathbf{v}}_k, \varphi \in V_k$ fixed and define $g : \mathbb{R} \longrightarrow \mathbb{R}$, that maps $u \longmapsto g_{m,k}(\hat{\mathbf{v}}_k + u\varphi) = \sum_{i=1}^{N} (\hat{p}_{k,i})^m d_\alpha^2\left(\tilde{\mathbf{x}}_{i,\hat{s}_{k,i}}, \hat{\mathbf{v}}_k + u\varphi\right)$. The function $g_{m,k}$ has a minimum in $\mathbf{v}_k = \hat{\mathbf{v}}_k$ if and only if g has a minimum in $u = 0$. A necessary condition is $g'(0) = 0$, that implies $\frac{dg_{m,k}}{du}(\hat{\mathbf{v}}_k + u\varphi)\Big|_{u=0} = 0$ for all $\varphi \in V_k$. For

every $\alpha \in [0,1)$, $d_\alpha(\cdot,\cdot)$ is the distance induced by the following inner product in V_k:

$$\langle \mathbf{y}_1, \mathbf{y}_2 \rangle_\alpha = \frac{1}{d} \sum_{\nu=1}^{d} \frac{w_\nu}{c_k} \int_0^{c_k} \left[(1-\alpha) y_1^{(\nu)}(t) y_2^{(\nu)}(t) + \alpha y_1'^{(\nu)}(t) y_2'^{(\nu)}(t) \right] \mathrm{d}t. \tag{S4}$$

For $\alpha = 1$, (S4) satisfies all the inner product properties in V_k, with the exception that $\langle \mathbf{y}, \mathbf{y} \rangle_1 = 0$ only implies $\mathbf{y} = \mathbf{const}$ a.e. in $(0, c_k)$; $d_1(\cdot,\cdot)$ is its induced semi-distance. Using this notation, we obtain

$$\begin{aligned}
0 &= \sum_{i=1}^{N} (\hat{p}_{k,i})^m \frac{d}{du} \left[\langle \tilde{\mathbf{x}}_{i,\hat{s}_{k,i}} - \hat{\mathbf{v}}_k - u\varphi, \tilde{\mathbf{x}}_{i,\hat{s}_{k,i}} - \hat{\mathbf{v}}_k - u\varphi \rangle_\alpha \right]_{u=0} \\
&= \sum_{i=1}^{N} (\hat{p}_{k,i})^m \frac{d}{du} \left[\langle \tilde{\mathbf{x}}_{i,\hat{s}_{k,i}} - \hat{\mathbf{v}}_k, \tilde{\mathbf{x}}_{i,\hat{s}_{k,i}} - \hat{\mathbf{v}}_k \rangle_\alpha \right. \\
&\qquad \left. -2u \langle \tilde{\mathbf{x}}_{i,\hat{s}_{k,i}} - \hat{\mathbf{v}}_k, \varphi \rangle_\alpha + u^2 \langle \varphi, \varphi \rangle_\alpha \right]_{u=0} \\
&= -2 \left\langle \sum_{i=1}^{N} (\hat{p}_{k,i})^m \left(\tilde{\mathbf{x}}_{i,\hat{s}_{k,i}} - \hat{\mathbf{v}}_k \right), \varphi \right\rangle_\alpha
\end{aligned}$$

for every $\varphi \in V_k$. In particular, we can choose $\varphi = \sum_{i=1}^{N} (\hat{p}_{k,i})^m \left(\tilde{\mathbf{x}}_{i,\hat{s}_{k,i}} - \hat{\mathbf{v}}_k \right)$, obtaining

$$\left\langle \sum_{i=1}^{N} (\hat{p}_{k,i})^m \left(\tilde{\mathbf{x}}_{i,\hat{s}_{k,i}} - \hat{\mathbf{v}}_k \right), \sum_{i=1}^{N} (\hat{p}_{k,i})^m \left(\tilde{\mathbf{x}}_{i,\hat{s}_{k,i}} - \hat{\mathbf{v}}_k \right) \right\rangle_\alpha = 0. \tag{S5}$$

If $\alpha \neq 1$, (S5) implies $\sum_{i=1}^{N} (\hat{p}_{k,i})^m \left(\tilde{\mathbf{x}}_{i,\hat{s}_{k,i}} - \hat{\mathbf{v}}_k \right) = \mathbf{0}$ a.e. in $(0, c_k)$. Using the non-degenerate clusters assumption $\sum_{i=1}^{N} \hat{p}_{k,i} > 0$ we obtain

$$\hat{\mathbf{v}}_k = \frac{\sum_{i=1}^{N} (\hat{p}_{k,i})^m \tilde{\mathbf{x}}_{i,\hat{s}_{k,i}}}{\sum_{i=1}^{N} (\hat{p}_{k,i})^m} \qquad \text{a.e. in } (0, c_k).$$

If $\alpha = 1$, (S5) implies $\sum_{i=1}^{N} (\hat{p}_{k,i})^m \left(\tilde{\mathbf{x}}_{i,\hat{s}_{k,i}} - \hat{\mathbf{v}}_k \right) = \mathbf{const}$ a.e. in $(0, c_k)$. Hence we have

$$\hat{\mathbf{v}}_k = \frac{\sum_{i=1}^{N} (\hat{p}_{k,i})^m \tilde{\mathbf{x}}_{i,\hat{s}_{k,i}}}{\sum_{i=1}^{N} (\hat{p}_{k,i})^m} + \mathbf{const} \qquad \text{a.e. in } (0, c_k).$$

To show the sufficiency of the previous equation for $\hat{\mathbf{v}}_k$ to be a minimizer, we compute the second-order derivative of g: $g''(u) = 2 \sum_{i=1}^{N} (\hat{p}_{k,i})^m \langle \varphi, \varphi \rangle_\alpha$. Since $\sum_{i=1}^{N} \hat{p}_{k,i} > 0$, we

have $g''(u) > 0$. Hence $u = 0$ is a strict local minimizer of g and $\hat{\mathbf{v}}_k$ is a strict local minimizer of $g_{m,k}$.

The constraint $\sum_{i=1}^{N} \hat{p}_{k,i} > 0$ implies also that the function $g_{m,k}$ is strictly convex on V_k, hence the local minimizer $\hat{\mathbf{v}}_k$ is actually the unique global minimizer of $g_{m,k}$. □

Proof of Lemma 3. The functions $t \mapsto t + s_{k,i}$, $y \mapsto y^m$, and d are continuous. The functional J_m is the sum of products of compositions of continuous functions, hence it is continuous. □

Proof of Theorem 4. Let $\mathbf{y}^{(iter-1)} = \left(\mathbf{P}^{(iter-1)}, \mathbf{S}^{(iter-1)}, \mathbf{v}_1^{(iter-1)}, \ldots, \mathbf{v}_K^{(iter-1)}\right) \in Y$. We have

$$\mathbf{y}^{(iter)} = \left(\mathbf{P}^{(iter)}, \mathbf{S}^{(iter)}, \mathbf{v}_1^{(iter)}, \ldots, \mathbf{v}_K^{(iter)}\right) \in T_m\left(\mathbf{y}^{(iter-1)}\right),$$

hence $\mathbf{v}_1^{(iter)}, \ldots, \mathbf{v}_K^{(iter)}$ are computed using equation (7). From Proposition 2, it follows that

$$J_m\left(\mathbf{P}^{(iter-1)}, \mathbf{S}^{(iter-1)}, \mathbf{v}_1^{(iter)}, \ldots, \mathbf{v}_K^{(iter)}\right) \leq J_m\left(\mathbf{P}^{(iter-1)}, \mathbf{S}^{(iter-1)}, \mathbf{v}_1^{(iter-1)}, \ldots, \mathbf{v}_K^{(iter-1)}\right).$$

Similarly, $\mathbf{S}^{(iter)}$ is selected in order to minimize the distances $d\left(\tilde{\mathbf{x}}_{i,s}, \mathbf{v}_k^{(iter)}\right)$, so

$$J_m\left(\mathbf{P}^{(iter-1)}, \mathbf{S}^{(iter)}, \mathbf{v}_1^{(iter)}, \ldots, \mathbf{v}_K^{(iter)}\right) \leq J_m\left(\mathbf{P}^{(iter-1)}, \mathbf{S}^{(iter-1)}, \mathbf{v}_1^{(iter)}, \ldots, \mathbf{v}_K^{(iter)}\right).$$

Finally, $\mathbf{P}^{(iter)}$ is computed using equations (3)-(4), hence from Preposition 1 we have

$$J_m\left(\mathbf{P}^{(iter)}, \mathbf{S}^{(iter)}, \mathbf{v}_1^{(iter)}, \ldots, \mathbf{v}_K^{(iter)}\right) \leq J_m\left(\mathbf{P}^{(iter-1)}, \mathbf{S}^{(iter)}, \mathbf{v}_1^{(iter)}, \ldots, \mathbf{v}_K^{(iter)}\right).$$

Combining the previous three equations we obtain (9).

Suppose now that $\mathbf{y}^{(iter-1)} \notin \Omega$. If (12) does not hold for $\mathbf{y}^{(iter-1)}$, then there exists $\mathbf{v}_k \in V_k$ with $\mathbf{v}_k \neq \mathbf{v}_k^{(iter-1)}$ such that

$$J_m\left(\mathbf{P}^{(iter-1)}, \mathbf{S}^{(iter-1)}, \mathbf{v}_1^{(iter-1)}, \ldots, \mathbf{v}_K^{(iter-1)}\right) \geq J_m\left(\mathbf{P}^{(iter-1)}, \mathbf{S}^{(iter-1)}, \mathbf{v}_1, \ldots, \mathbf{v}_K\right)$$

and by Proposition 2 we know that $J_m(\mathbf{P}^{(iter-1)}, \mathbf{S}^{(iter-1)}, \cdot)$ has a unique global minimizer, hence

$$J_m\left(\mathbf{P}^{(iter-1)}, \mathbf{S}^{(iter-1)}, \mathbf{v}_1^{(iter)}, \ldots, \mathbf{v}_K^{(iter)}\right) < J_m\left(\mathbf{P}^{(iter-1)}, \mathbf{S}^{(iter-1)}, \mathbf{v}_1^{(iter-1)}, \ldots, \mathbf{v}_K^{(iter-1)}\right).$$

If $\mathbf{y}^{(iter-1)}$ satisfies (12) but not (11), then $\mathbf{v}_k^{(iter)} = \mathbf{v}_k^{(iter-1)}$ and there exists $\mathbf{S} \in \mathbb{R}^{K\times N}$ such that

$$J_m\left(\mathbf{P}^{(iter-1)}, \mathbf{S}^{(iter-1)}, \mathbf{v}_1^{(iter)}, \ldots, \mathbf{v}_K^{(iter)}\right) > J_m\left(\mathbf{P}^{(iter-1)}, \mathbf{S}, \mathbf{v}_1^{(iter)}, \ldots, \mathbf{v}_K^{(iter)}\right).$$

Since $\mathbf{S}^{(iter)}$ minimizes $d\left(\tilde{\mathbf{x}}_{i,s}, \mathbf{v}_k^{(iter)}\right)$ we have that

$$J_m\left(\mathbf{P}^{(iter-1)}, \mathbf{S}^{(iter)}, \mathbf{v}_1^{(iter)}, \ldots, \mathbf{v}_K^{(iter)}\right) < J_m\left(\mathbf{P}^{(iter-1)}, \mathbf{S}^{(iter-1)}, \mathbf{v}_1^{(iter)}, \ldots, \mathbf{v}_K^{(iter)}\right).$$

Finally, if $\mathbf{y}^{(iter-1)}$ satisfies (11) and (12), but not (10), we have that $\mathbf{v}_k^{(iter)} = \mathbf{v}_k^{(iter-1)}$ and there exists $\mathbf{P} \in [0,1]^{K\times N}$, with $\sum_{k=1}^K p_{k,i} = 1$ and $\sum_{i=1}^N p_{k,i} > 0$, such that

$$J_m\left(\mathbf{P}^{(iter-1)}, \mathbf{S}^{(iter-1)}, \mathbf{v}_1^{(iter)}, \ldots, \mathbf{v}_K^{(iter)}\right) > J_m\left(\mathbf{P}, \mathbf{S}^{(iter-1)}, \mathbf{v}_1^{(iter)}, \ldots, \mathbf{v}_K^{(iter)}\right).$$

Then from Proposition 1 and equation (11) follows that

$$\begin{aligned} J_m\left(\mathbf{P}^{(iter)}, \mathbf{S}^{(iter-1)}, \mathbf{v}_1^{(iter)}, \ldots, \mathbf{v}_K^{(iter)}\right) &< J_m\left(\mathbf{P}^{(iter-1)}, \mathbf{S}^{(iter-1)}, \mathbf{v}_1^{(iter)}, \ldots, \mathbf{v}_K^{(iter)}\right) \\ &\leq J_m\left(\mathbf{P}^{(iter-1)}, \mathbf{S}^{(iter)}, \mathbf{v}_1^{(iter)}, \ldots, \mathbf{v}_K^{(iter)}\right). \end{aligned}$$

The matrix $\mathbf{S}^{(iter)}$ minimizes $d\left(\tilde{\mathbf{x}}_{i,s}, \mathbf{v}_k^{(iter)}\right)$ and does not depend on $\mathbf{P}$, so we have

$$J_m\left(\mathbf{P}^{(iter)}, \mathbf{S}^{(iter)}, \mathbf{v}_1^{(iter)}, \ldots, \mathbf{v}_K^{(iter)}\right) \leq J_m\left(\mathbf{P}^{(iter)}, \mathbf{S}^{(iter-1)}, \mathbf{v}_1^{(iter)}, \ldots, \mathbf{v}_K^{(iter)}\right).$$

As a consequence

$$J_m\left(\mathbf{P}^{(iter)}, \mathbf{S}^{(iter)}, \mathbf{v}_1^{(iter)}, \ldots, \mathbf{v}_K^{(iter)}\right) < J_m\left(\mathbf{P}^{(iter-1)}, \mathbf{S}^{(iter)}, \mathbf{v}_1^{(iter)}, \ldots, \mathbf{v}_K^{(iter)}\right).$$

□

S2 Implementation details of probKMA

S2.1 Cluster cleaning and detection of portions belonging to each cluster

ProbKMA lacks the ability to distinguish the case of a curve $\mathbf{x}_{i_1}$ that matches *all* K cluster centers (i.e. $d(\tilde{\mathbf{x}}_{i_1}, \mathbf{v}_k) = \epsilon$, $\forall k$, with $\epsilon \approx 0$) from that of a curve $\mathbf{x}_{i_2}$ that does *not* match *any* of the K cluster centers (i.e. $d(\tilde{\mathbf{x}}_{i_2}, \mathbf{v}_k) = M$, $\forall k$, with $M \gg 0$). In both cases equation (3) leads to the same membership probabilities $p_{k,i_1} = p_{k,i_2} \approx 1/K$. To overcome this issue, we perform a *cluster cleaning* step when the algorithm is near convergence. The membership matrix $\mathbf{P}$ is dichotomized, that is all membership probabilities are transformed into either 0 or 1. We consider the quantile $q_{\frac{1}{K}}$ of order $\frac{1}{K}$ of all distances $d(\tilde{\mathbf{x}}_{i,s_{k,i}}, \mathbf{v}_k)$; membership probabilities $p_{k,i}$ corresponding to distances lower than $q_{\frac{1}{K}}$ are set to 1, while all others are set to 0. Note that, in this way, each curve $\mathbf{x}_i$ is allowed not to belong to any cluster, as well as to more than one cluster. This step distinguishes among the two extreme cases above (as long as all or most curves are not extreme cases), leading to clean membership probabilities $p_{k,i_1} = 1$ and $p_{k,i_2} = 0$, $k = 1, \ldots, K$. Note that the order of the quantile employed in the dichotomization can be changed based on our expectations or on the distribution of all distances $d(\tilde{\mathbf{x}}_i, \mathbf{v}_k)$. For example, if we expect many curves to belong to no cluster, we can decrease the order of the quantile to set more memberships to 0.

This cleaning step is performed also at the end of the algorithm, to detect the curve portions belonging to each cluster, i.e. the ones whose dichotomized membership is equal to 1, and to employ them to obtain better estimates of the cluster centers through equation (7). These "clean" results are then employed in the computation of the generalized silhouette index (see Section 3) and in the functional motif discovery post-processing (see Sections 3 and S3).

S2.2 Dealing with large gaps in the curves

As shown in Subsection 2.1, from a theoretical point of view the input curves are required to satisfy only mild regularity conditions that are typical in functional data analysis. In

practice, as for almost all methods in functional data analysis, probKMA works best if the curves are reasonably smooth. Curve smoothing can address this need and highly improve results. Moreover, in real applications each functional datum must be created from discrete evaluations – possibly available on datum-specific and/or irregular grids, with some measurements missing relative to other data. This too is tackled with smoothing, and other straightforward pre-processing steps to fill small gaps. However, in some applications input curves present also *large gaps*, i.e. miss entire subregions that cannot be meaningfully imputed by smoothing (see, for instance, our application to mutagenesis data in Section 5). Functional methods that consider the curves globally are not appropriate for this type of data. On the contrary, our method can tolerate large gaps because it exploits the functional data locally. We formalize this situation allowing the i^{th} input curve $\mathbf{x}_i : \ D_i \longrightarrow \mathbb{R}^d$ to have a domain $D_i \subseteq \mathbb{R}$ defined as a finite union of intervals. The distance in (5) is thus generalized as

$$\begin{aligned} \tilde{d}_\alpha^2(\mathbf{x}, \mathbf{v}) = \sum_{\nu=1}^{d} \frac{w_\nu}{d} \Bigg[& \frac{1-\alpha}{|(0,c) \cap D|} \int_{(0,c)\cap D} \left(x^{(\nu)}(t) - v^{(\nu)}(t)\right)^2 \mathrm{d}t \\ & + \frac{\alpha}{|(0,c) \cap D|} \int_{(0,c)\cap D} \left(x'^{(\nu)}(t) - v'^{(\nu)}(t)\right)^2 \mathrm{d}t \Bigg], \end{aligned} \tag{S6}$$

where D is the domain of $\mathbf{x}$. While the distance in (5) is computed on the whole interval $(0, c)$ where the cluster center $\mathbf{v}$ is defined, the dissimilarity in (S6) is computed only on the portion of $(0, c)$ that intersects the domain of $\mathbf{x}$, and is well defined only if this intersection is not empty, i.e. $|(0, c) \cap D| > 0$. Based on (S6), the equation to update the k-th cluster center becomes

$$\hat{\mathbf{v}}_k = \frac{\displaystyle\sum_{i=1}^{N} \frac{(\hat{p}_{k,i})^m}{|(0, c_k) \cap \tilde{D}_{i,s_{k,i}}|} \mathbb{I}_{(0,c_k)\cap \tilde{D}_{i,\hat{s}_{k,i}}} \tilde{\mathbf{x}}_{i,\hat{s}_{k,i}}}{\displaystyle\sum_{i=1}^{N} \frac{(\hat{p}_{k,i})^m}{|(0, c_k) \cap \tilde{D}_{i,s_{k,i}}|} \mathbb{I}_{(0,c_k)\cap \tilde{D}_{i,\hat{s}_{k,i}}}} \tag{S7}$$

a.e. on $(0, c_k) \cap (\bigcup_{i=1}^{N} \tilde{D}_{i,\hat{s}_{k,i}})$. Here $\mathbb{I}_A$ is the indicator function of the set A, and $\tilde{D}_{i,s} = D_i - s$ is the domain of the shifted curve $\tilde{\mathbf{x}}_{i,s}$. For $\alpha = 1$, $\hat{\mathbf{v}}_k$ is defined up to an additive constant.

Note that equations (S7) and (7) are very similar: the k^{th} cluster center is still a weighted average of the shifted curves, but the contribution of a curve now depends on its domain,

in addition to its probability of belonging to cluster k. When no large gaps are present in the input curves, (S7) reduces to (7).

S3 Functional motif discovery post-processing

Given the set of candidate motifs – obtained from multiple runs of probKMA and filtered based on generalized silhouette indices and number of occurrences – we need a post-processing to merge similar candidate motifs and locate all instances of the final set of functional motifs (see Section 3). Merging and motif search can be tackled with many different strategies, each with its pros and cons. To save computation, we devised implementations that take advantage of the large amount of information gained from the multiple runs of probKMA – in particular, of the distances that have been computed between each candidate motif and all input curves (both the ones that contain the motif, and the ones that do not). Briefly, for merging we group similar candidate motifs benchmarking their pairwise distance against the distances between all motifs and curves. In each group, we select one representative motif based on its number of occurrences and the average distance between the motif and its occurrences in the curves. If the candidate motifs in a group have very different lengths, we can select more than one representative. This allows us to retain both a long motif and a shorter motif – which may have a larger number of occurrences, or a smaller average distance. In addition, we estimate the variability within each group of motifs, and we use it to define the radius R used in the motif search step. This is group-specific and it allows us to map instances of each motif based on a data-driven evaluation of its own signal-to-noise ratio.

In detail, we propose the following implementation for the post-processing.

1. Compute all pairwise distances between candidate motifs;

2. Perform hierarchical clustering with average linkage of candidate motifs, using their pairwise distances;

3. Determine a global radius R_{all} based on the minimum distances between all candidate motifs and all curves;

4. Cut the hierarchical clustering dendrogram at height $2R_{all}$, obtaining M groups of similar motifs;

5. For each group $m = 1, \ldots, M$:

   a. Determine a group-specific radius R_m based on the minimum distances between the motifs of group m and all curves;

   b. For each motif in group m

      - Find the curves containing the motif, i.e. the curves with distance $\leq R_m$ from the motif;
      - Approximate the number of occurrences in the curves (portions of curves with distance $\leq R_m$ from the motif), counting the number of curves containing the motif;
      - Approximate the average within-motif distance (i.e. the average distance between the motif and all its occurrences in the curves) with the average of the minimum distances between the motif and each of the curves containing it;

   c. Select a very small number of motifs based on the approximate number of occurrences, the approximate average within-motif distance and the motif length;

6. Find all occurrences of the selected motifs (portions of curves with distance $\leq R_m$ from the motif).

Pairwise distances between candidate motifs in step 1 are computed according to the same distance employed by probKMA, allowing alignment between each pair of motifs but requiring a minimum overlap, defined as a percentage of the shortest motif in each pair (default choice 60%).

During the last iteration of probKMA algorithm (and in particular in step iii) we compute the minimum distances between each motif and all curves. Moreover, the curves are divided in two groups: the ones that contain the motif, and the ones that do not contain it. In steps 3 and 5a of the post-processing we utilize this piece of information in order to compute the radii R_{all} and R_m (see Fig. S1). In particular, we employ k-nearest neighbors

to select the distance that best discriminate the group of distances between each motif and the curves that contain it (group 1), from the group of distances between each motif and the curves that do not contain it (group 0). This distance is selected based on the posterior probability of the k-nearest neighbors classifier (i.e. the percentage of votes for one group). The advantage of using a non-parametric classifier such as k-nearest neighbors is that we do not need any assumption on the distribution of distances in the two groups. However, it is important to observe that the resolution of the posterior depends on the number of neighbors used ($k = 1$ corresponds to probability of 0 and 1, $k = 2$ to 0, 0.5 and 1 and so on). In addition, the posterior probability of the k-nearest neighbors classifier can be not decreasing: for intermediate distances between the two groups we can have a distance classified as group 0, then a distance classified as group 1, and then again a distance classified as group 0. In order to be conservative in defining group 1, we select the smallest distance at which the posterior probability of belonging to group 1 is smaller than a given threshold.

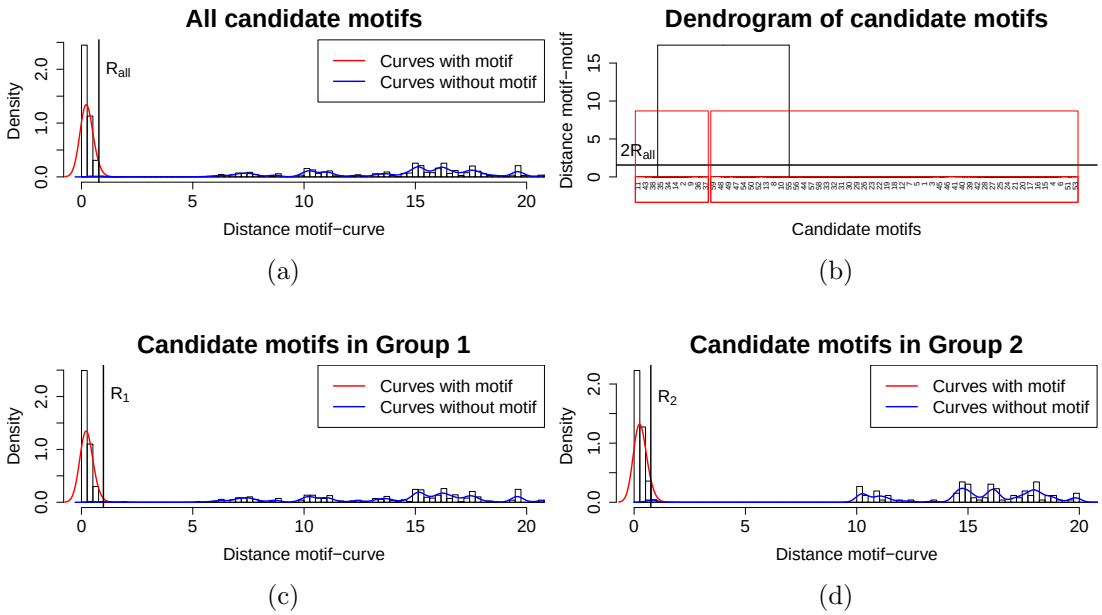


Figure S1: Example of post-processing, data from first simulation scenario in Subsection 4.2, $l = 200$, $\sigma = 1$. (a) Selection of a global radius R_{all} (step 3); (b) Hierarchical clustering dendrogram cut at height $2R_{all}$ (step 4); (c)-(d) Selection of a group-specific radius R_m in each of the two groups (step 5a).

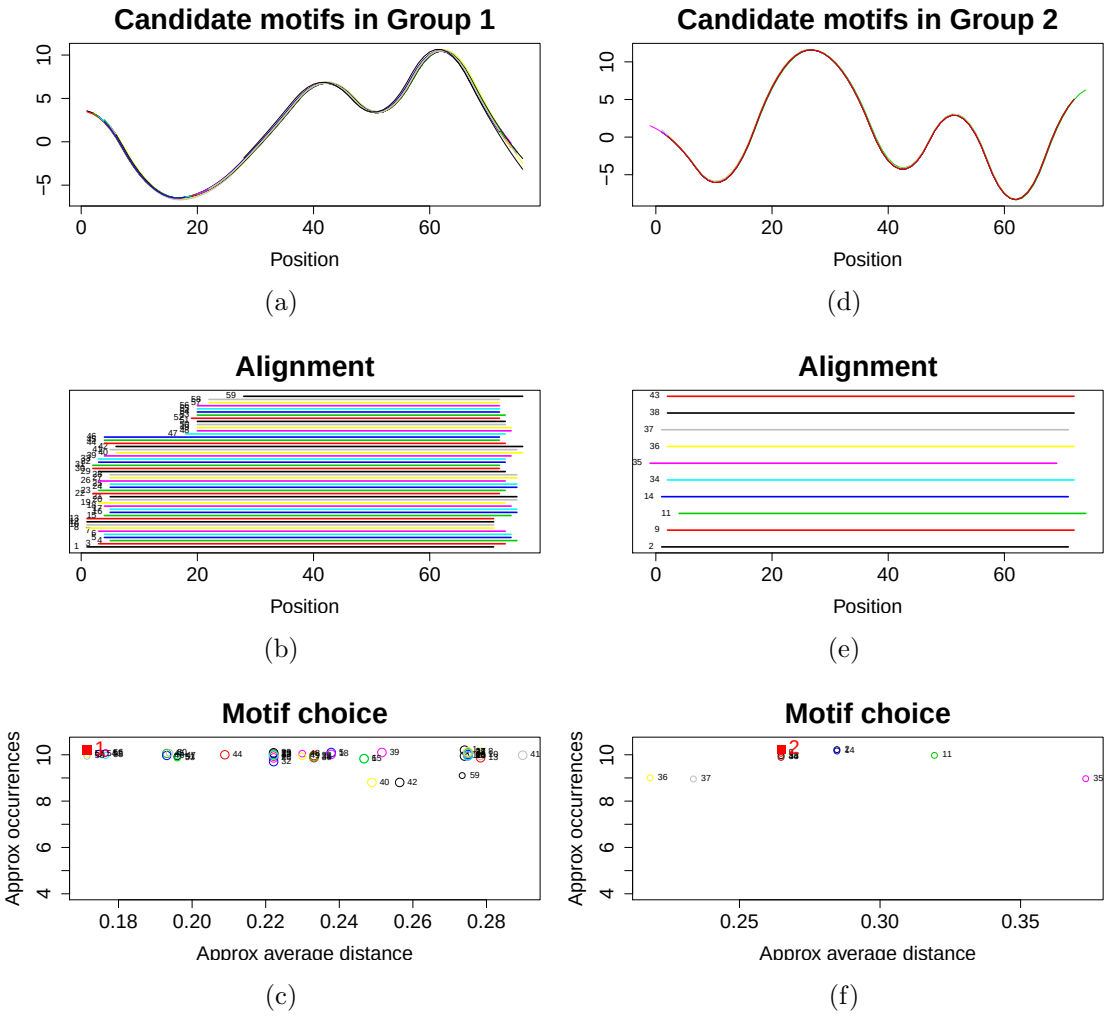


Figure S2: Example of post-processing, data from first simulation scenario in Subsection 4.2, $l = 200$, $\sigma = 1$. (a) Aligned candidate motifs in Group 1; (b) Alignment of candidate motifs in Group 1; (c) Approximate average within-motif distance vs approximate number of occurrences, for motifs in Group 1 (step 5b). Circle size is proportional to motif length. A red square indicates selected motif (step 5c); (d)-(f) Analogous plots for motifs in Group 2.

Regarding the selection of k, with very small k the algorithm is very fast but give very noisy results, while larger k create more stable results but take much more computational time. Our default choice is $k = 3$ and threshold 0.5 for the posterior probability.

Computing approximate number of occurrences and approximate average within-motif distance in step 5b allows us to avoid finding all occurrences of all motifs (i.e. all portions

of curves with distance $\leq R_m$ from each motif m) – which would be computationally expensive. This approximation is done considering, for each motif, only one occurrence in the curves containing it. If the motif occurs multiple times in a curve, only the occurrence closest to the motif is considered. As a consequence, the approximate metrics represent lower bounds of the actual metrics. Note that this approximation is good when only a small number of curves contain multiple instances of the same motif. Our implementation lets the user choose between the approximate metrics and the actual ones (the default choice is approximate metrics, which is computationally lighter).

Selection of motifs in each group m (step 5c, Fig. S2) is done maximizing the approximate number of occurrences while simultaneously minimizing the approximate average within-motif distance. In particular, we order the motifs based on the sum of their ranks in each of the two dimensions, and we select the top one (see Figs. S2c and S2f). If other motifs in the same group have length very different from the selected one, we can allow the procedure to select them too (by default, we select only the top motif in each group).

Step 6 represents a motif search step, that is needed in order to locate all occurrences of the selected motifs, i.e. all portions of curves with distance $\leq R_m$ from each motif (with R_m the group-specific radius for that motif). We observe that two overlapping portions of curves tend to be quite similar. Hence, if a portion of curve matches a particular motif, the portion of curve that begins and ends immediately at its left/right is likely to match the same motif too. In order to avoid counting multiple times the same motif occurrence, we require that two occurrences of the same motif are well separated (analogously to Lin et al., 2002). In particular, let $\mathbf{v}_m$ be a selected motif that matches a curve $\mathbf{x}$ in two portions corresponding to the shifts $s_1 < s_2$, i.e. $d\left(\tilde{\mathbf{x}}_{s_1}, \mathbf{v}_m\right) \leq R_m$ and $d\left(\tilde{\mathbf{x}}_{s_2}, \mathbf{v}_m\right) \leq R_m$. In order to count both occurrences, we ask that there exists a shift $s \in (s_1, s_2)$ such that $d\left(\tilde{\mathbf{x}}_s, \mathbf{v}_m\right) > R_m$. Among all the $s \in (s_l, s_r)$ such that $d\left(\tilde{\mathbf{x}}_s, \mathbf{v}_m\right) \leq R_m$, with $d\left(\tilde{\mathbf{x}}_{s_l}, \mathbf{v}_m\right) > R_m$ and $d\left(\tilde{\mathbf{x}}_{s_r}, \mathbf{v}_m\right) > R_m$, we select the shift that minimizes $d\left(\tilde{\mathbf{x}}_s, \mathbf{v}_m\right)$.

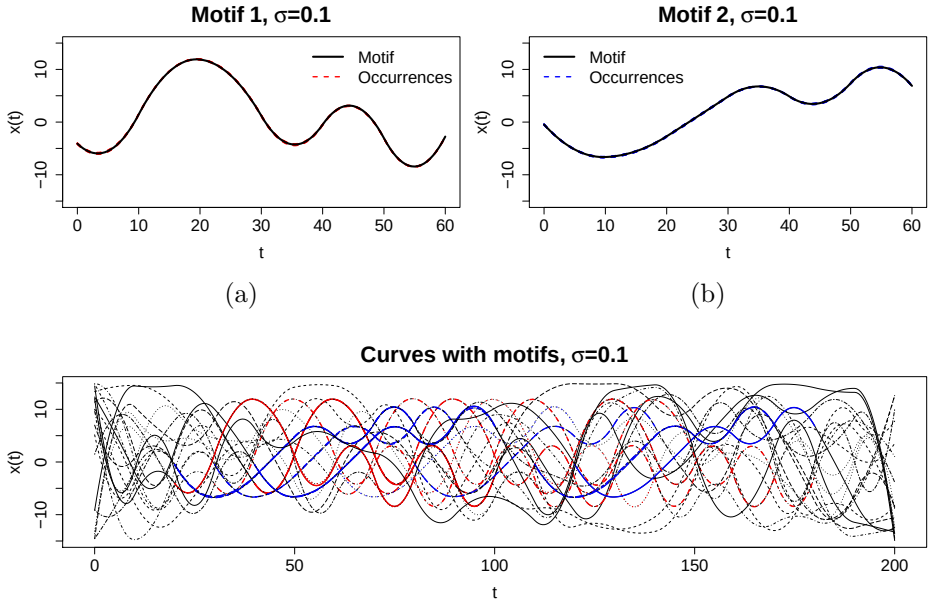
S4 Simulations: additional results

S4.1 Functional motif discovery: varying curve length and noise in motifs

Here we report additional information related to simulations in Subsection 4.2.

Figs. S3-S5 show the two functional motifs, the 12 aligned occurrences of each motif, and the 20 curves embedding occurrences of the two motifs, for curve length $\ell = 200$ and noise levels $\sigma = 0.1, 0.5, 2$ in simulation scenario (1) (data for $\sigma = 1$ are shown in Fig. 2). Figs. S6-S8 show the performance of our probKMA-based functional motif discovery for different levels of noise and curve lengths $\ell = 300, 400, 500$ (results for curve length $\ell = 200$ are shown in Fig. 3). Analogous information for simulation scenario (2) are showed in Figs. S9-S16.

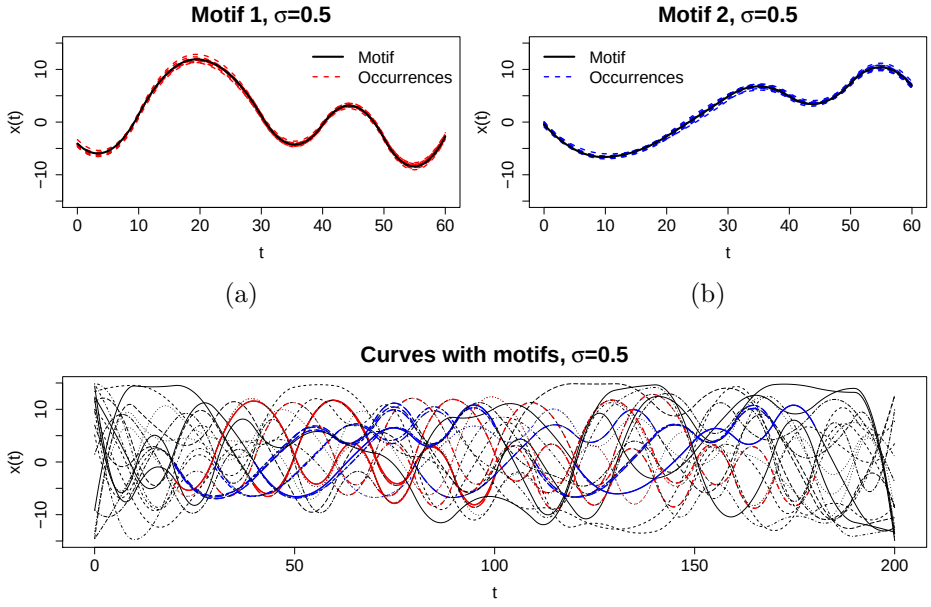
In simulation scenario (1), we employ Sobolev-like distance $d_{0.5}(\cdot, \cdot)$ to measure similarities between portions of curves, while in simulation scenario (2) we use the L^2-like pseudo-distance $d_1(\cdot, \cdot)$ on the weak derivative. In both cases, probKMA is run for $K = 2, 3$, minimum motif lengths $c_{min} = 40, 50, 60$, and 20 random initializations for each (K, c_{min}) pair. The same initializations (i.e. the same initial membership matrix $\mathbf{P}^{(0)}$ and shift matrix $\mathbf{S}^{(0)}$) are employed for all ℓ and σ combinations. The weighting fuzziness parameter is fixed to be $m = 2$. ProbKMA iterations are stopped when the global Bhattacharyya distance $BC_{\max} = \max_{k=1,\dots,K} BC_k$ is $\leq 10^{-8}$. The maximum motif length is set to 70. Elongation step is performed at every iteration, when $BC_{\max} \leq 10^{-3}$; each center is elongated up to 50% of its length in either directions, requiring that the relative objective function $J_{m,k}$ increase is less than 5% (i.e. $(J_{m,k,\text{elong}} - J_{m,k})/J_{m,k} < 0.05$). Cleaning step is performed every 50 iterations, when $BC_{\max} \leq 10^{-4}$. Candidate motifs that belong to less than 5 curves, as well as the ones with an average cluster silhouette index lower than the 90^{th} percentile of all overall average silhouette indices, are filtered out (see Section 3). Post-processing is performed with default values (see Section S3).



(a) (b)

(c)

Figure S3: Simulation scenario (1) with $\ell = 200$ and $\sigma = 0.1$. (a), (b) Two functional motifs (black solid curves) and 12 aligned occurrences of each (red and blue dashed curves); (c) 20 curves embedding occurrences of the two motifs (red and blue portions, respectively).



(a) (b)

(c)

Figure S4: Simulation scenario (1) with $\ell = 200$ and $\sigma = 0.5$. (a), (b) Two functional motifs (black solid curves) and 12 aligned occurrences of each (red and blue dashed curves); (c) 20 curves embedding occurrences of the two motifs (red and blue portions, respectively).

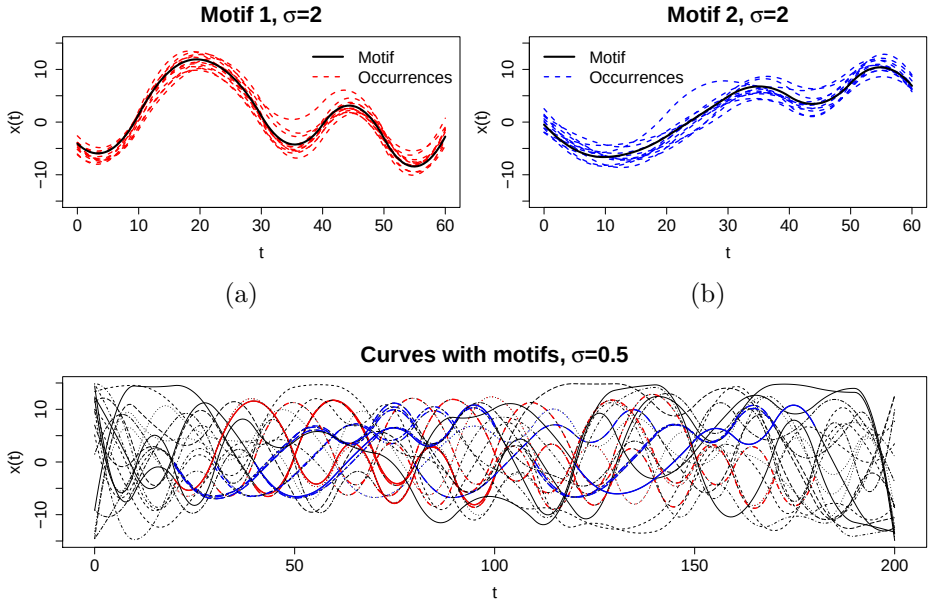
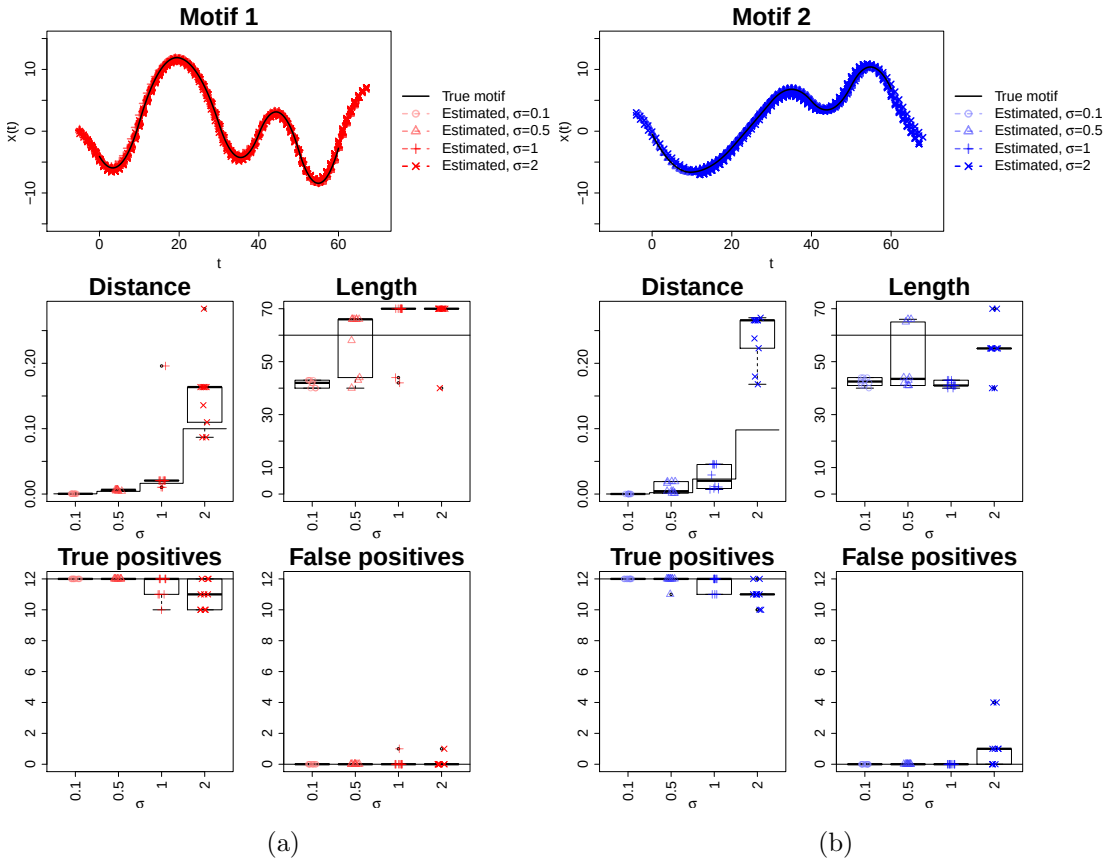


Figure S5: Simulation scenario (1) with $\ell = 200$ and $\sigma = 2$. (a), (b) Two functional motifs (black solid curves) and 12 aligned occurrences of each (red and blue dashed curves); (c) 20 curves embedding occurrences of the two motifs (red and blue portions, respectively).



(a) (b)

Figure S6: Functional motif discovery results for simulation scenario (1) with $\ell = 300$ and various levels of σ. (a) Motif 1; (b) Motif 2. The boxplots in the lower half of the panels are obtained from 10 replications at each σ value. In 33 cases, exactly 2 motifs are found; in the remaining 7 cases, one additional motif is discovered.

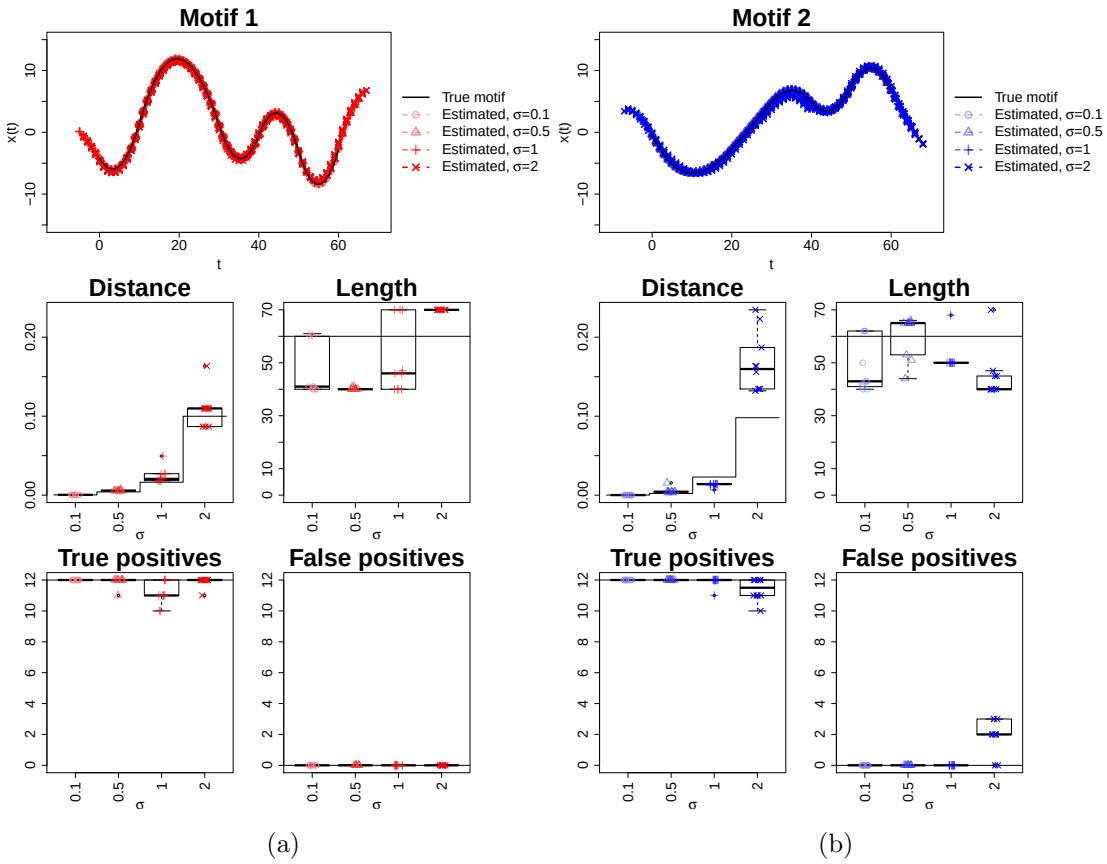
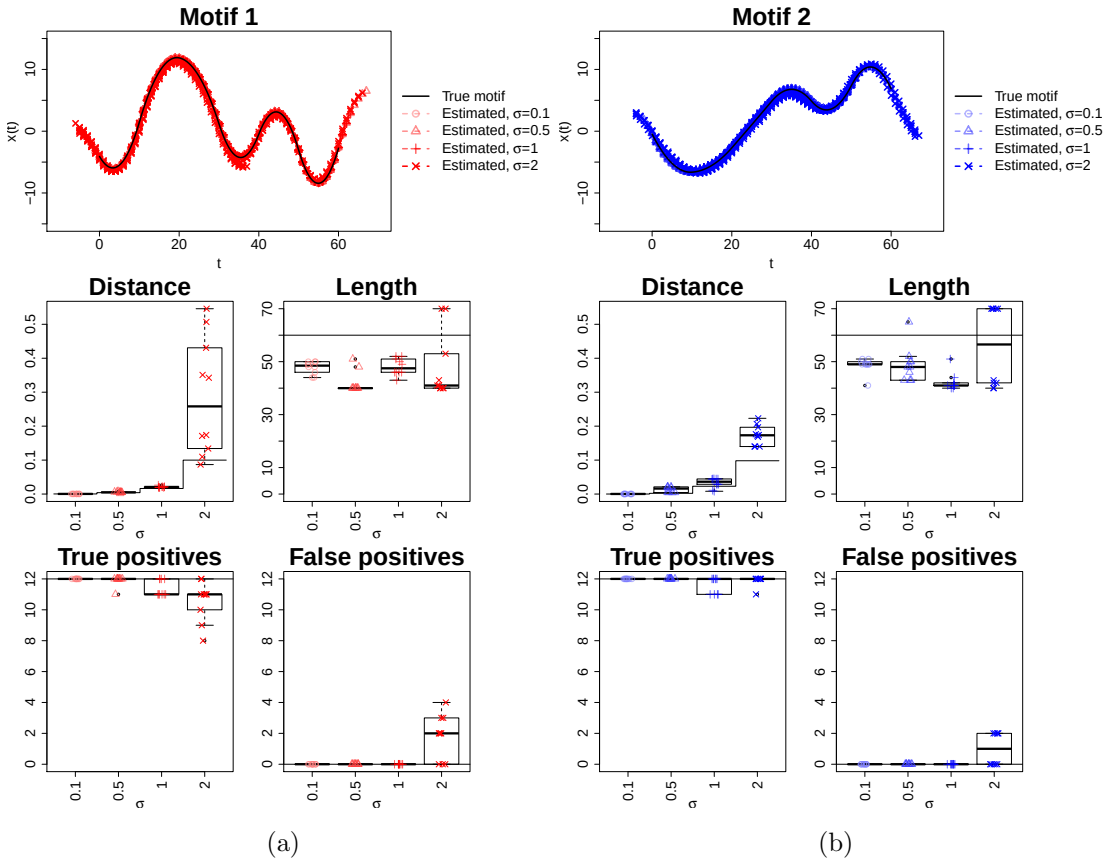
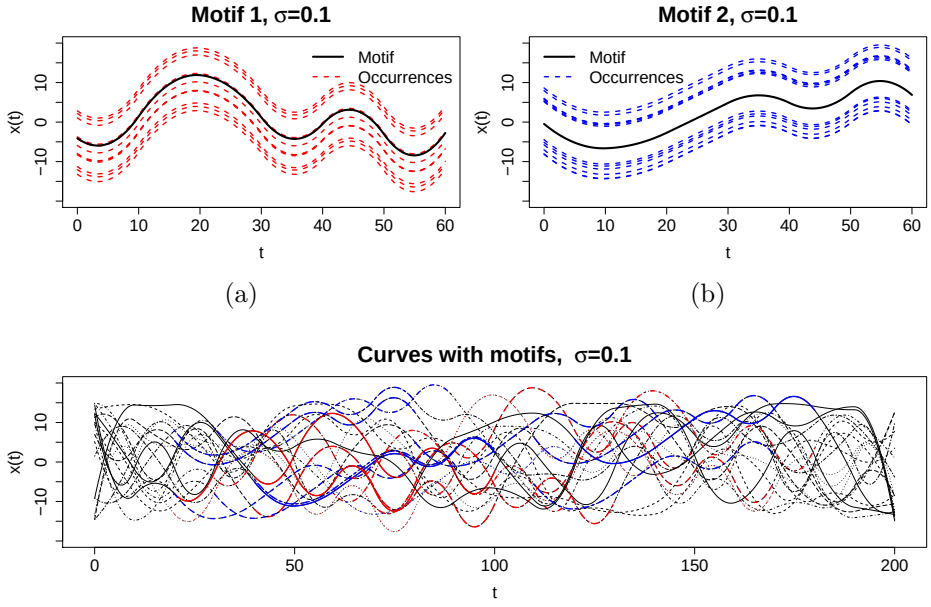


Figure S7: Functional motif discovery results for simulation scenario (1) with $\ell = 400$ and various levels of σ. (a) Motif 1; (b) Motif 2. The boxplots in the lower half of the panels are obtained from 10 replications at each σ value. For all the considered noise levels, exactly 2 motifs are found.



(a) (b)

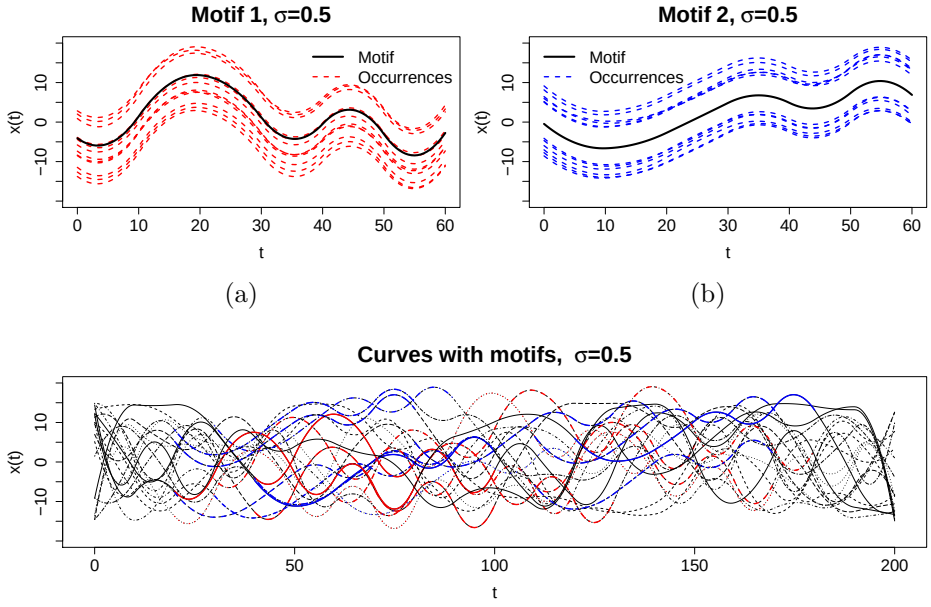
Figure S8: Functional motif discovery results for simulation scenario (1) with $\ell = 500$ and various levels of σ. (a) Motif 1; (b) Motif 2. The boxplots in the lower half of the panels are obtained from 10 replications at each σ value. In 34 cases, exactly 2 motifs are found; in the remaining 6 cases, one additional motif is discovered.



(a) (b)

(c)

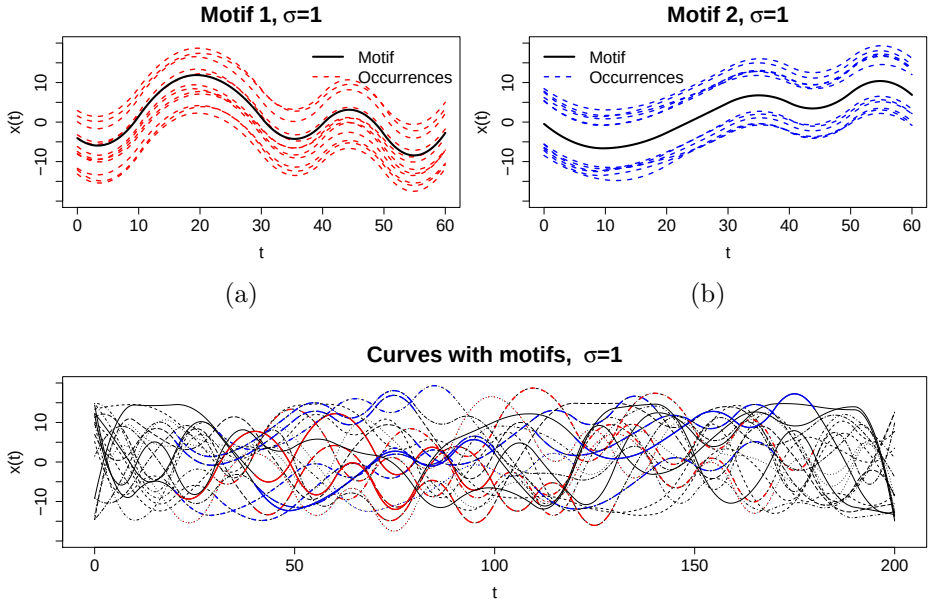
Figure S9: Simulation scenario (2) with $\ell = 200$ and $\sigma = 0.1$. (a), (b) Two functional motifs (black solid curves) and 12 aligned occurrences of each (red and blue dashed curves); (c) 20 curves embedding occurrences of the two motifs (red and blue portions, respectively).



(a) (b)

(c)

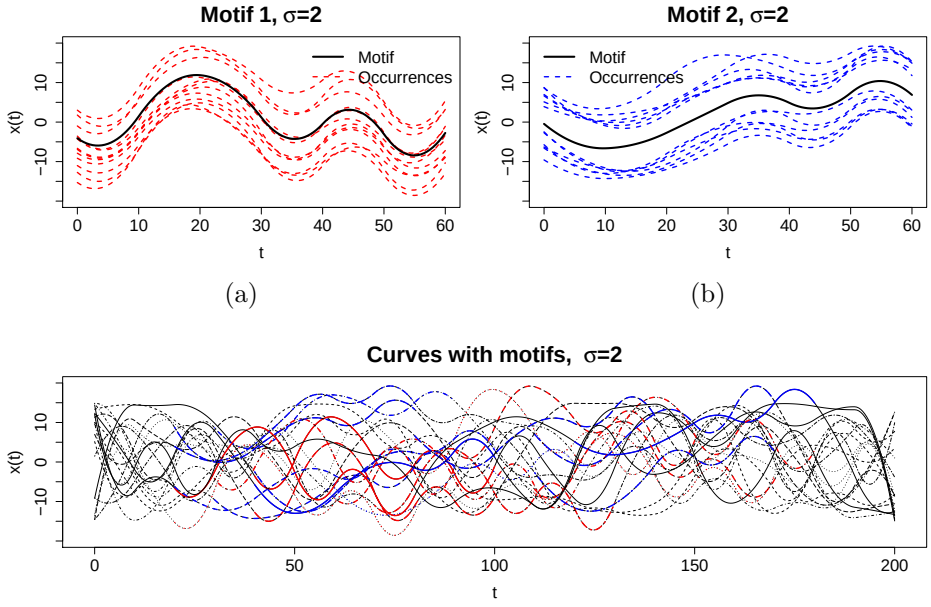
Figure S10: Simulation scenario (2) with $\ell = 200$ and $\sigma = 0.5$. (a), (b) Two functional motifs (black solid curves) and 12 aligned occurrences of each (red and blue dashed curves); (c) 20 curves embedding occurrences of the two motifs (red and blue portions, respectively).



(a) (b)

(c)

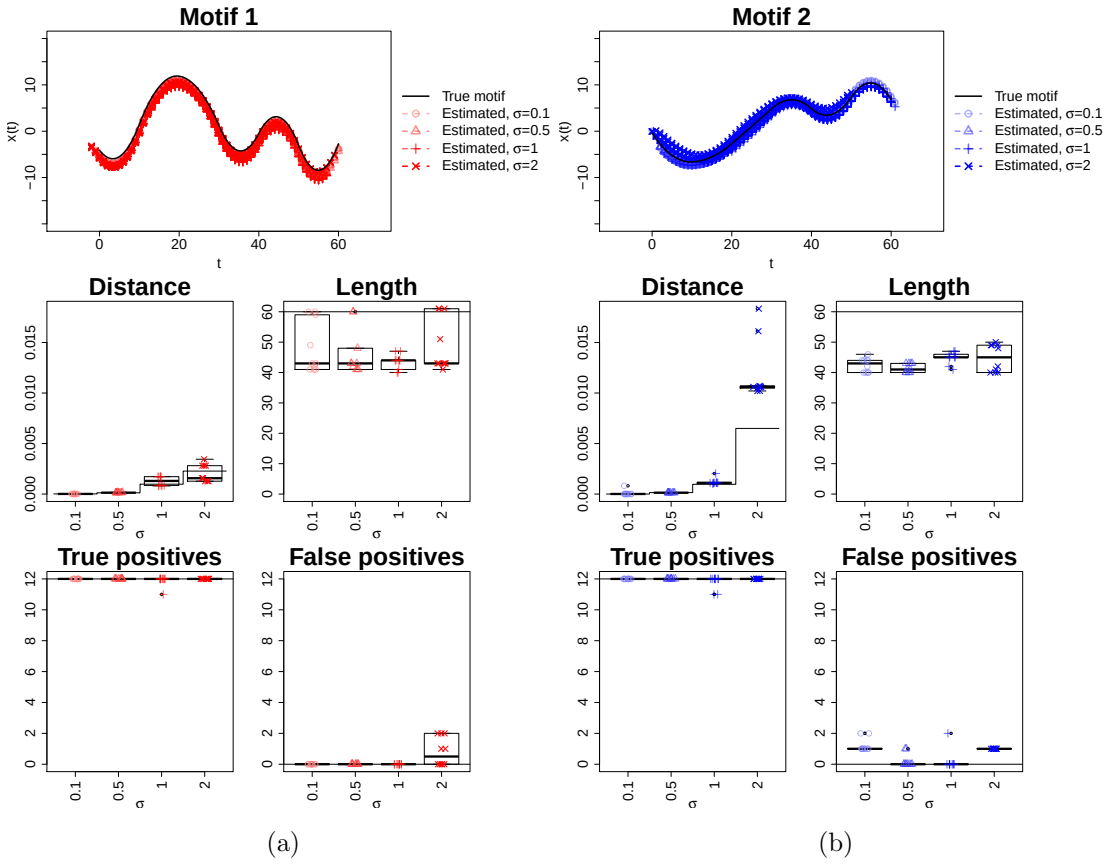
Figure S11: Simulation scenario (2) with $\ell = 200$ and $\sigma = 1$. (a), (b) Two functional motifs (black solid curves) and 12 aligned occurrences of each (red and blue dashed curves); (c) 20 curves embedding occurrences of the two motifs (red and blue portions, respectively).



(a) (b)

(c)

Figure S12: Simulation scenario (2) with $\ell = 200$ and $\sigma = 2$. (a), (b) Two functional motifs (black solid curves) and 12 aligned occurrences of each (red and blue dashed curves); (c) 20 curves embedding occurrences of the two motifs (red and blue portions, respectively).



(a) (b)

Figure S13: Functional motif discovery results for simulation scenario (2) with $\ell = 200$ and various levels of σ. (a) Motif 1; (b) Motif 2. The boxplots in the lower half of the panels are obtained from 10 replications at each σ value. In 34 cases, exactly 2 motifs are found; in 5 cases, one additional motif is discovered; in 1 case, two additional motifs are discovered.

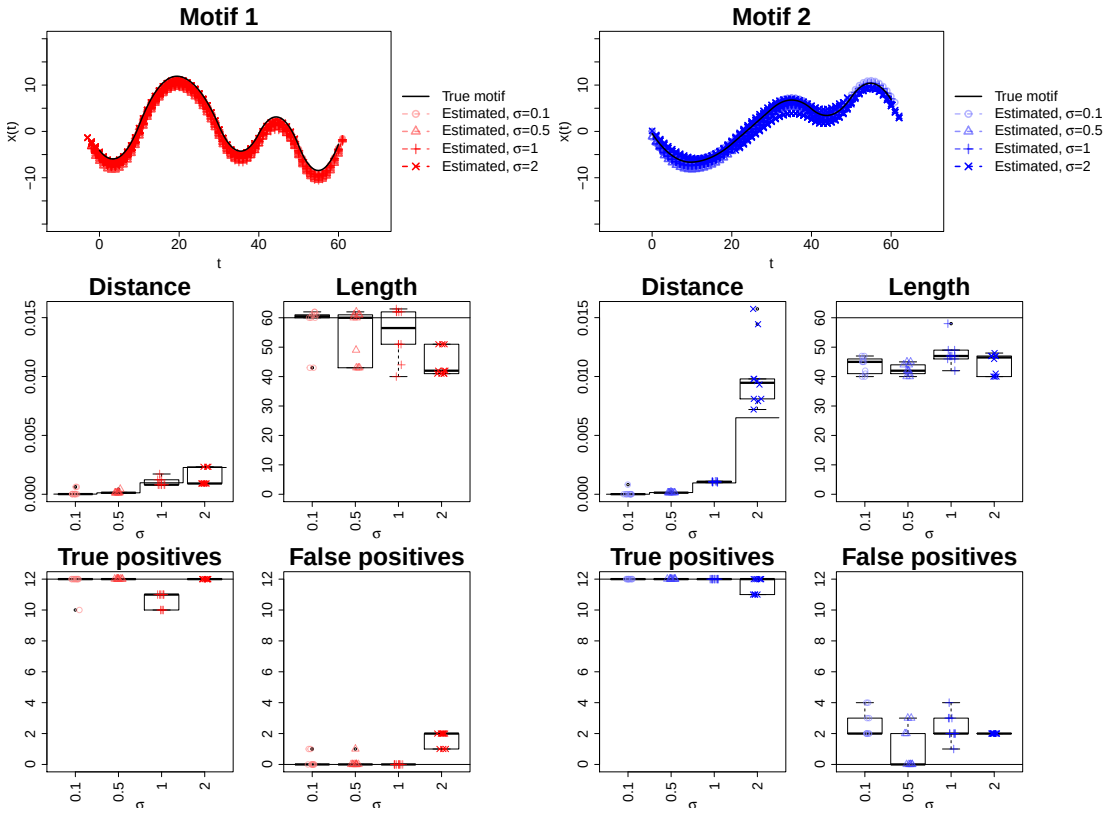


Figure S14: Functional motif discovery results for simulation scenario (2) with $\ell = 300$ and various levels of σ. (a) Motif 1; (b) Motif 2. The boxplots in the lower half of the panels are obtained from 10 replications at each σ value. In 16 cases, exactly 2 motifs are found; in 13 cases, one additional motif is discovered; in 11 cases, at least 2 additional motifs are discovered.

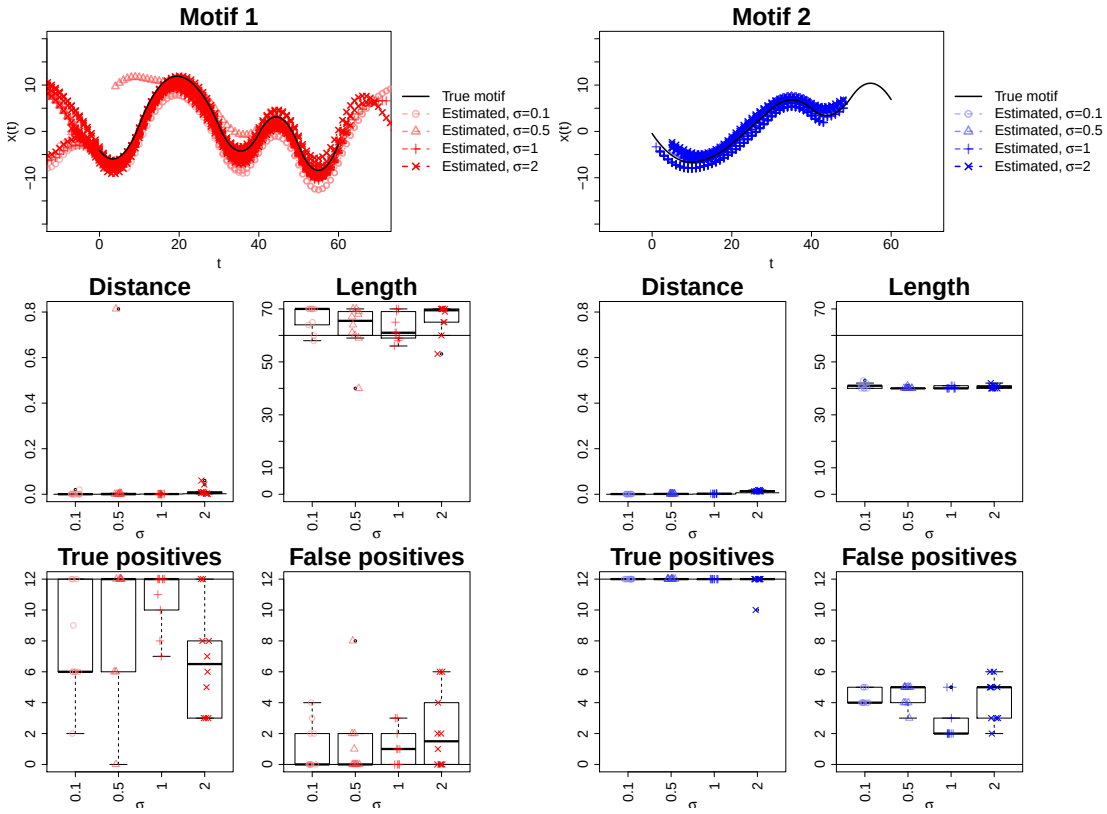


Figure S15: Functional motif discovery results for simulation scenario (2) with $\ell = 400$ and various levels of σ. (a) Motif 1; (b) Motif 2. The boxplots in the lower half of the panels are obtained from 10 replications at each σ value. In 2 cases, exactly 2 motifs are found; in 18 cases, one additional motif is discovered; in 20 cases, at least 2 additional motifs are discovered.

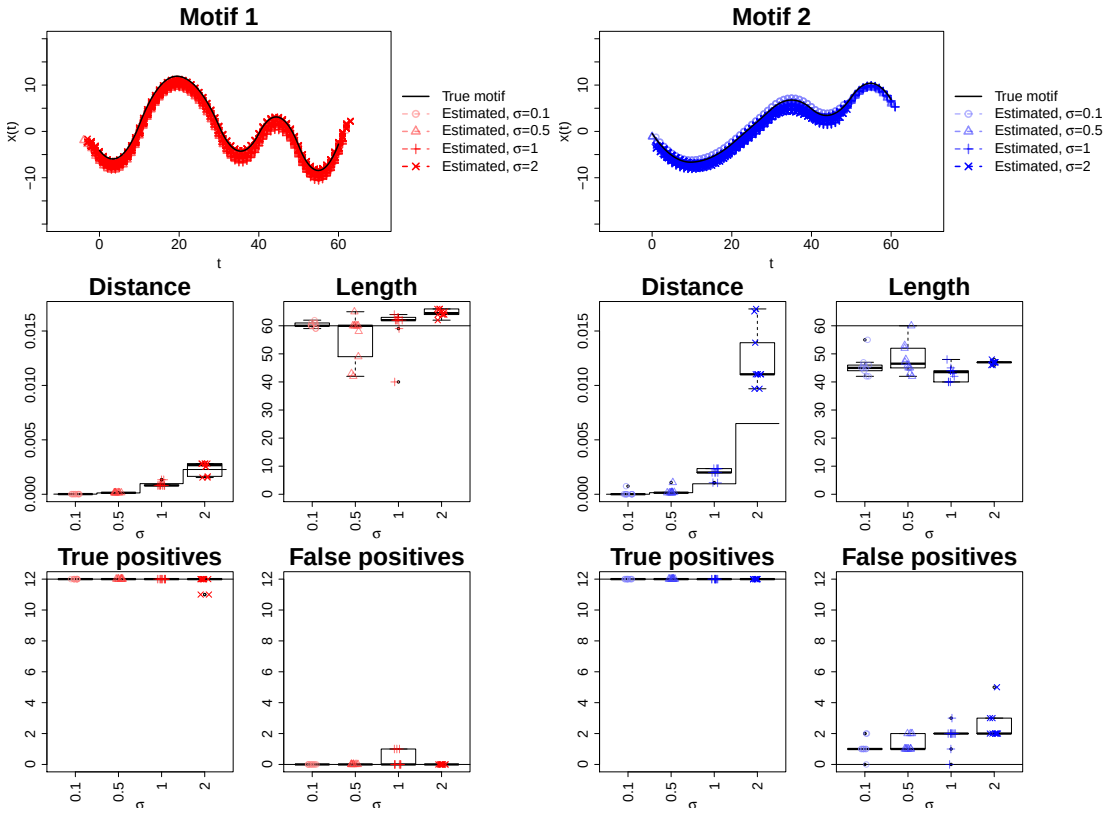
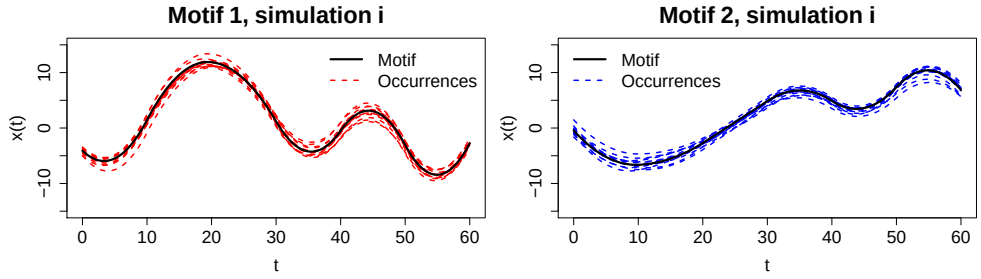


Figure S16: Functional motif discovery results for simulation scenario (2) with $\ell = 500$ and various levels of σ. (a) Motif 1; (b) Motif 2. The boxplots in the lower half of the panels are obtained from 10 replications at each σ value. In 16 cases, exactly 2 motifs are found; in 13 cases, one additional motif is discovered; in 25 cases, at least 2 additional motifs are discovered.

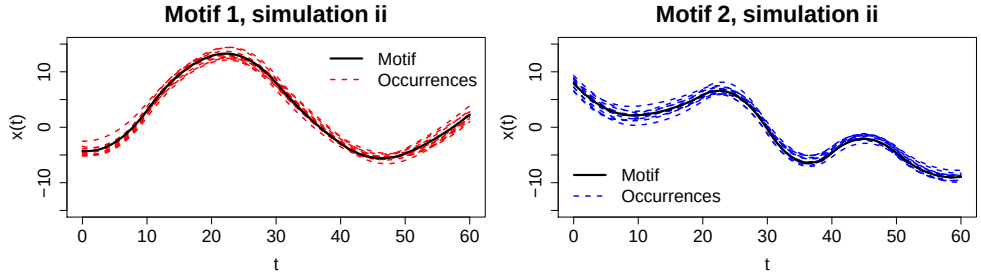
To validate the results described in Subsection 4.2 and in the previous figures, we repeat simulations in both scenarios 10 times, considering 10 different randomly generated pairs of motifs and re-generating the curves' background. Fig. S17 shows aligned occurrences of functional motifs (for curve length $\ell = 200$ and level of noise $\sigma = 1$), for the 10 different pairs of functional motifs and set of curves in simulation scenario (1). Summary results of 10 replications of functional motif discovery for each of these 10 datasets are shown in Fig. S18. Figs. S19-S20 show the corresponding plots in simulation scenario (2). In all cases, our method shows good performance and similar behaviors as curve length and noise level change. This is evidence that its effectiveness does not depend on the specific shapes of the motifs embedded in the curves. Moreover, these results corroborate the observations we previously made. When the level of noise increases, the distance between true and estimated motifs increases, true positives decreases, and false positive increases. These performance measures are only slightly affected by an increase in curve length (and hence in the background/motif ratio). Neither curve length nor noise level affects the motif length, which the method usually underestimates. Fig. S21 shows the number of motifs discovered in the 10 replications of functional motif discovery, for the 10 simulations in both scenarios. Curve length affects the number of motifs discovered: when the background/motif ratio increases, our method tends to discover more than two motifs in the curves.


Motif 1, simulation i
Motif
Occurrences
x(t)
t

(a)

Motif 2, simulation i
Motif
Occurrences
x(t)
t

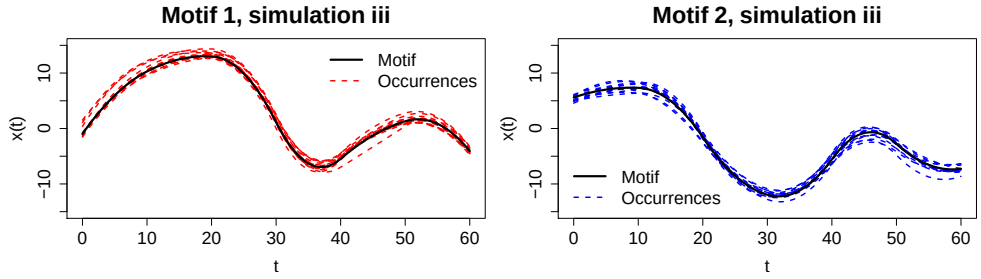
(b)


Motif 1, simulation ii
Motif
Occurrences
x(t)
t

(c)

Motif 2, simulation ii
Motif
Occurrences
x(t)
t

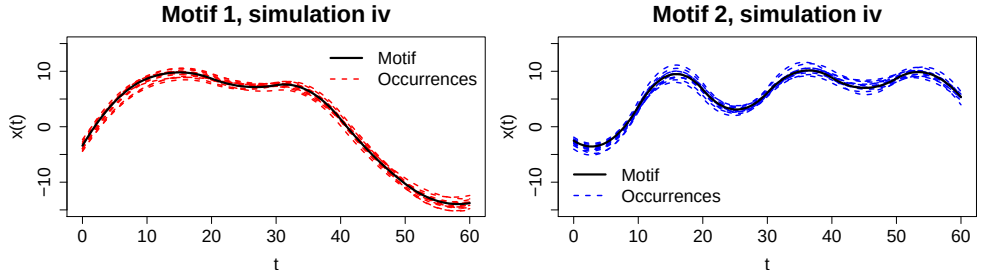
(d)


Motif 1, simulation iii
Motif
Occurrences
x(t)
t

(e)

Motif 2, simulation iii
Motif
Occurrences
x(t)
t

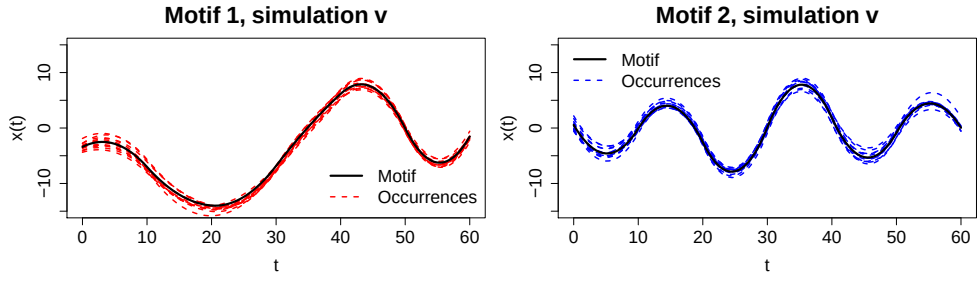
(f)


Motif 1, simulation iv
Motif
Occurrences
x(t)
t

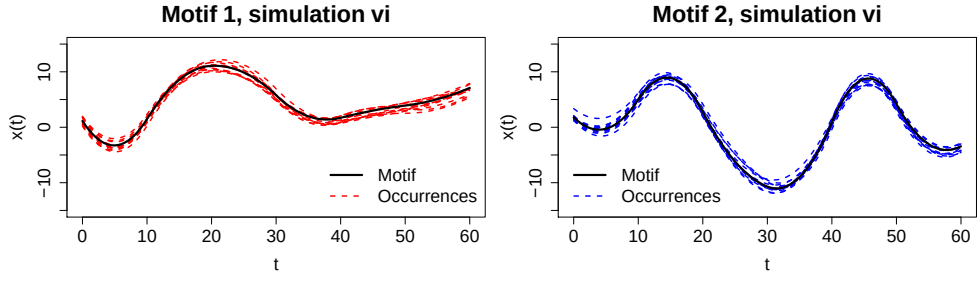
(g)

Motif 2, simulation iv
Motif
Occurrences
x(t)
t

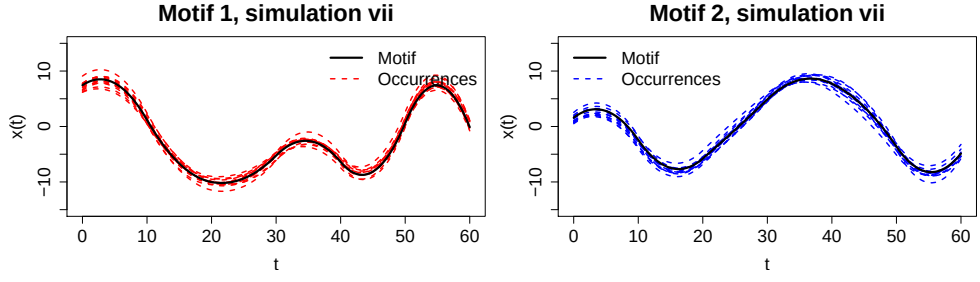
(h)


Motif 1, simulation v
Motif 2, simulation v
x(t)
t
Motif
Occurrences

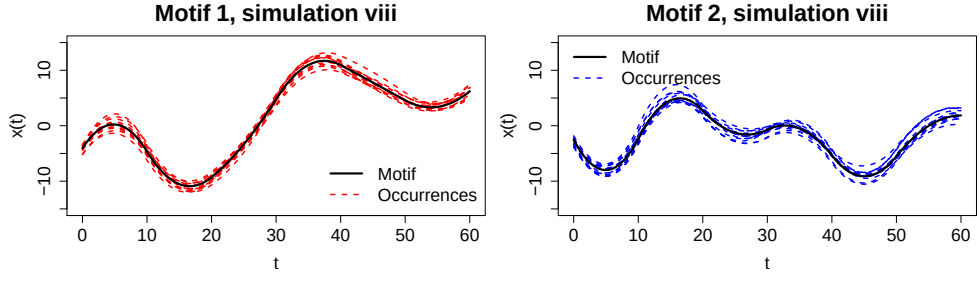
(i) (j)


Motif 1, simulation vi
Motif 2, simulation vi
x(t)
t
Motif
Occurrences

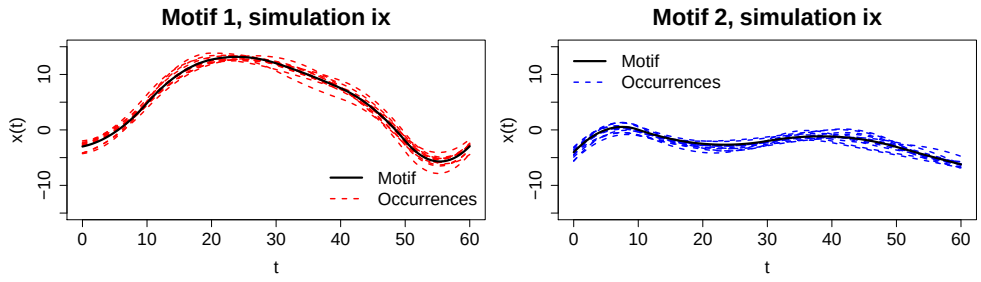
(k) (l)


Motif 1, simulation vii
Motif 2, simulation vii
x(t)
t
Motif
Occurrences

(m) (n)


Motif 1, simulation viii
Motif 2, simulation viii
x(t)
t
Motif
Occurrences

(o) (p)



(q) (r)

(s) (t)

Figure S17: The two functional motifs (black solid curves) and the 12 aligned occurrences of each (red and blue dashed curves), for 10 different datasets in simulation scenario (1), for $\ell = 200$ and $\sigma = 1$.

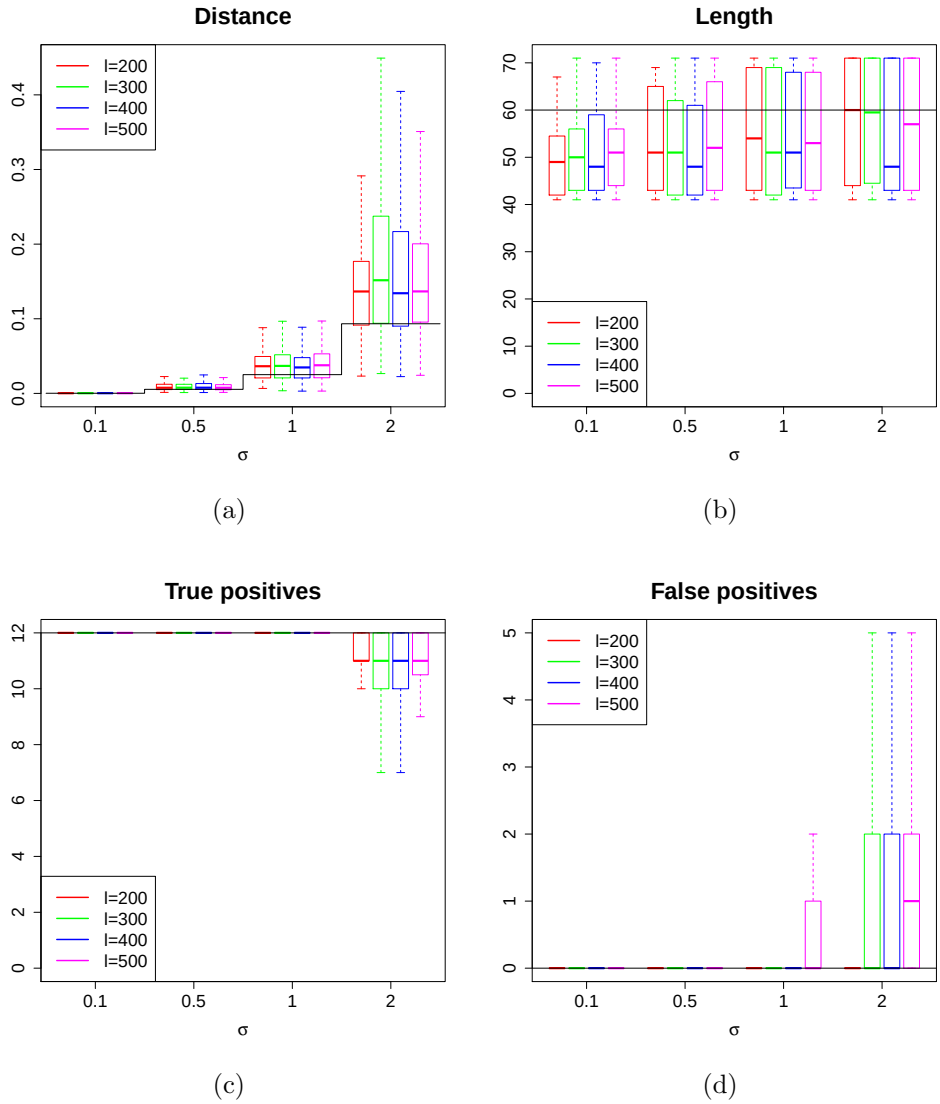
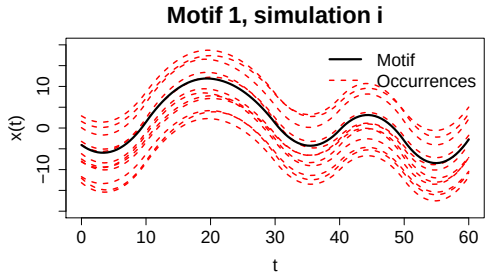
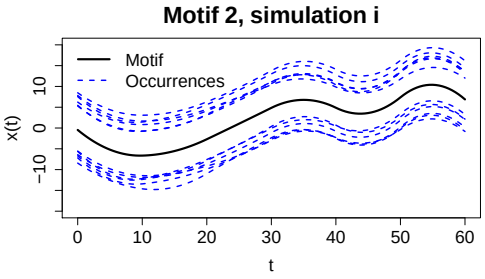


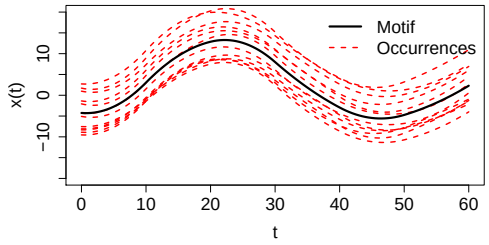
Figure S18: Summary of functional motif discovery results for the 10 different datasets in simulation scenario (1). (a) Distance between true and estimated motifs; (b) Estimated length of motifs; (c) Number of true positives; (d) Number of false positives. The boxplots are obtained from 10 replications at each of the 10 different datasets, and both motifs (a total of 200 observations). Outliers are not plotted, for clarity of visualization.


Motif 1, simulation i
Motif
Occurrences
x(t)
t

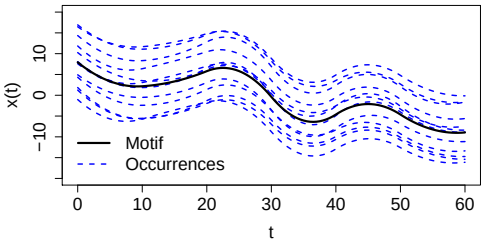
(a)


Motif 2, simulation i
Motif
Occurrences
x(t)
t

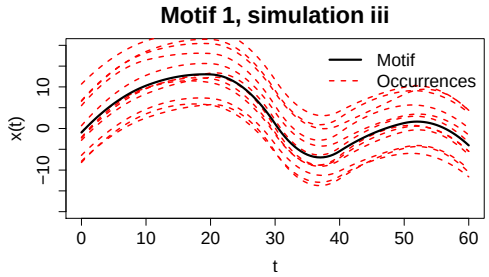
(b)


Motif 1, simulation ii
Motif
Occurrences
x(t)
t

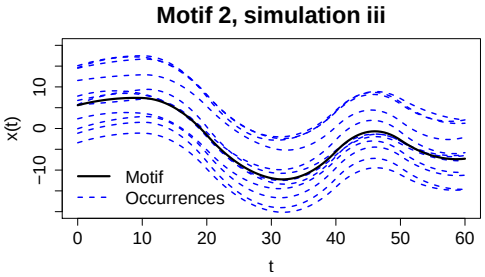
(c)


Motif 2, simulation ii
Motif
Occurrences
x(t)
t

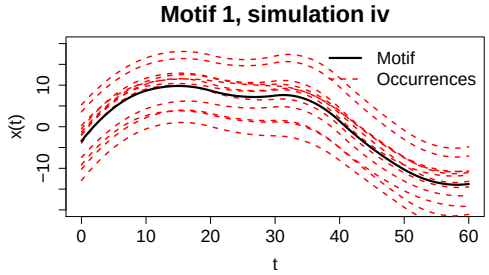
(d)


Motif 1, simulation iii
Motif
Occurrences
x(t)
t

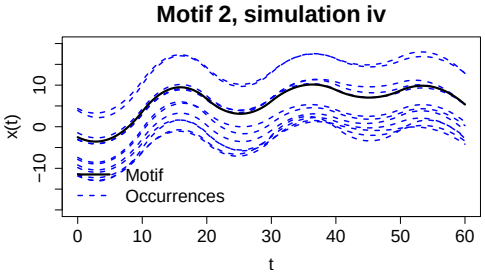
(e)


Motif 2, simulation iii
Motif
Occurrences
x(t)
t

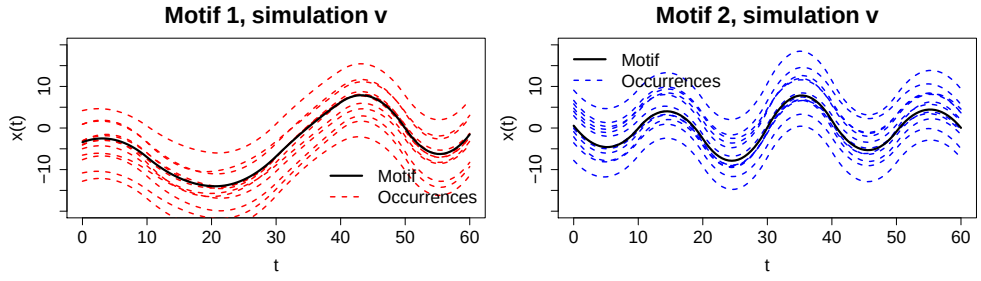
(f)


Motif 1, simulation iv
Motif
Occurrences
x(t)
t

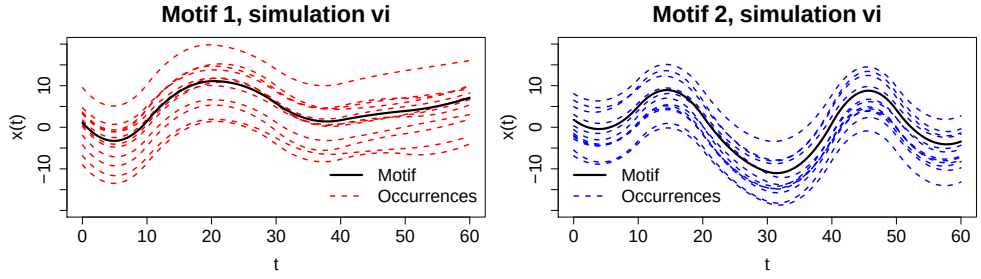
(g)


Motif 2, simulation iv
Motif
Occurrences
x(t)
t

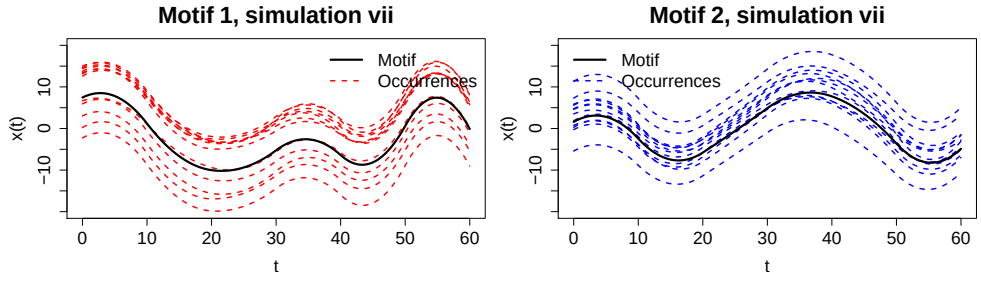
(h)


Motif 1, simulation v
Motif 2, simulation v
x(t)
t
Motif
Occurrences

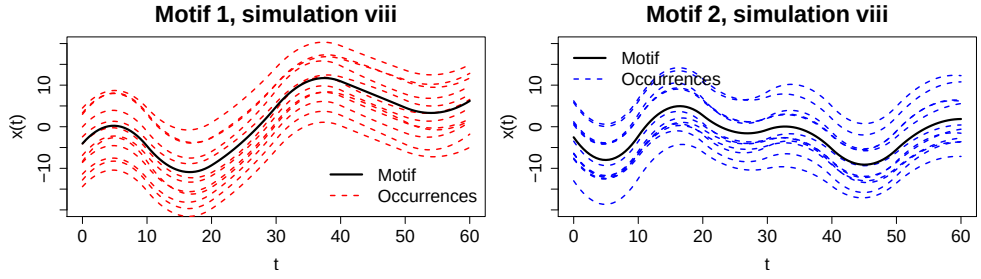
(i) (j)


Motif 1, simulation vi
Motif 2, simulation vi
x(t)
t
Motif
Occurrences

(k) (l)


Motif 1, simulation vii
Motif 2, simulation vii
x(t)
t
Motif
Occurrences

(m) (n)


Motif 1, simulation viii
Motif 2, simulation viii
x(t)
t
Motif
Occurrences

(o) (p)

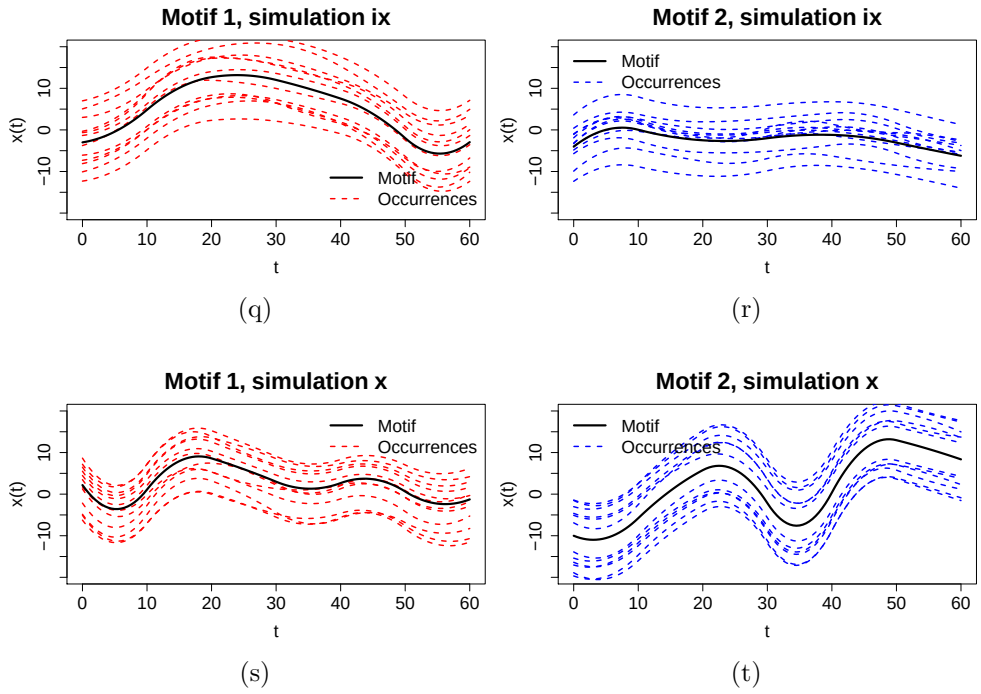


Figure S19: The two functional motifs (black solid curves) and the 12 aligned occurrences of each (red and blue dashed curves), for 10 different datasets in simulation scenario (2), for $\ell = 200$ and $\sigma = 1$.

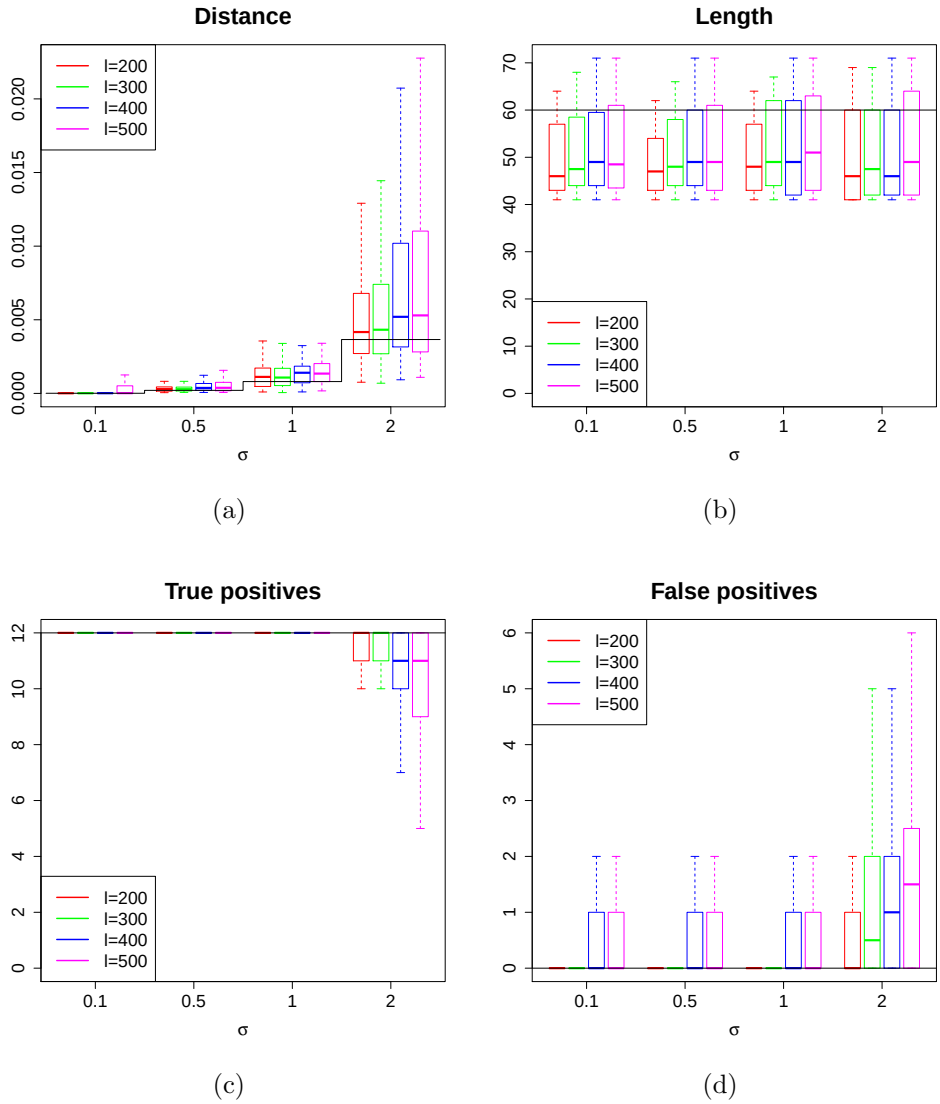
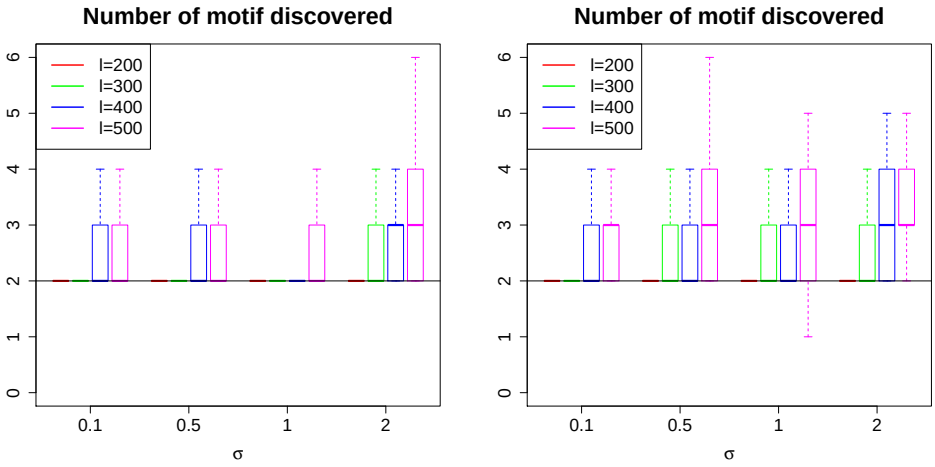


Figure S20: Summary of functional motif discovery results for the 10 different datasets in simulation scenario (2). (a) Distance between true and estimated motifs; (b) Estimated length of motifs; (c) Number of true positives; (d) Number of false positives. The boxplots are obtained from 10 replications at each of the 10 different datasets, and both motifs (a total of 200 observations). Outliers are not plotted, for clarity of visualization.



(a) (b)

Figure S21: Number of motifs discovered by functional motif discovery method for the 10 different datasets in (a) simulation scenario (1) and (b) simulation scenario (2). The boxplots are obtained from 10 replications at each of the 10 different datasets (a total of 100 observations). Outliers are not plotted, for clarity of visualization.

Finally, we consider the 10 simulations for the first scenario and examine how results change with the number of random initializations used to run probKMA for each (K, c) pair. To do this, we subsample the probKMA runs from the analysis already conducted, which employed 20 random initializations, using only 5, 10, or 15 initializations for each (K, c_{min}) pair. We re-run the post-processing steps and compare results with those previously obtained with all 20 initializations. Reassuringly, our method is robust to the number of initializations employed, and retains its good performance even when probKMA is run only 5 times for each choice of (K, c_{min}) (see Fig. S22).

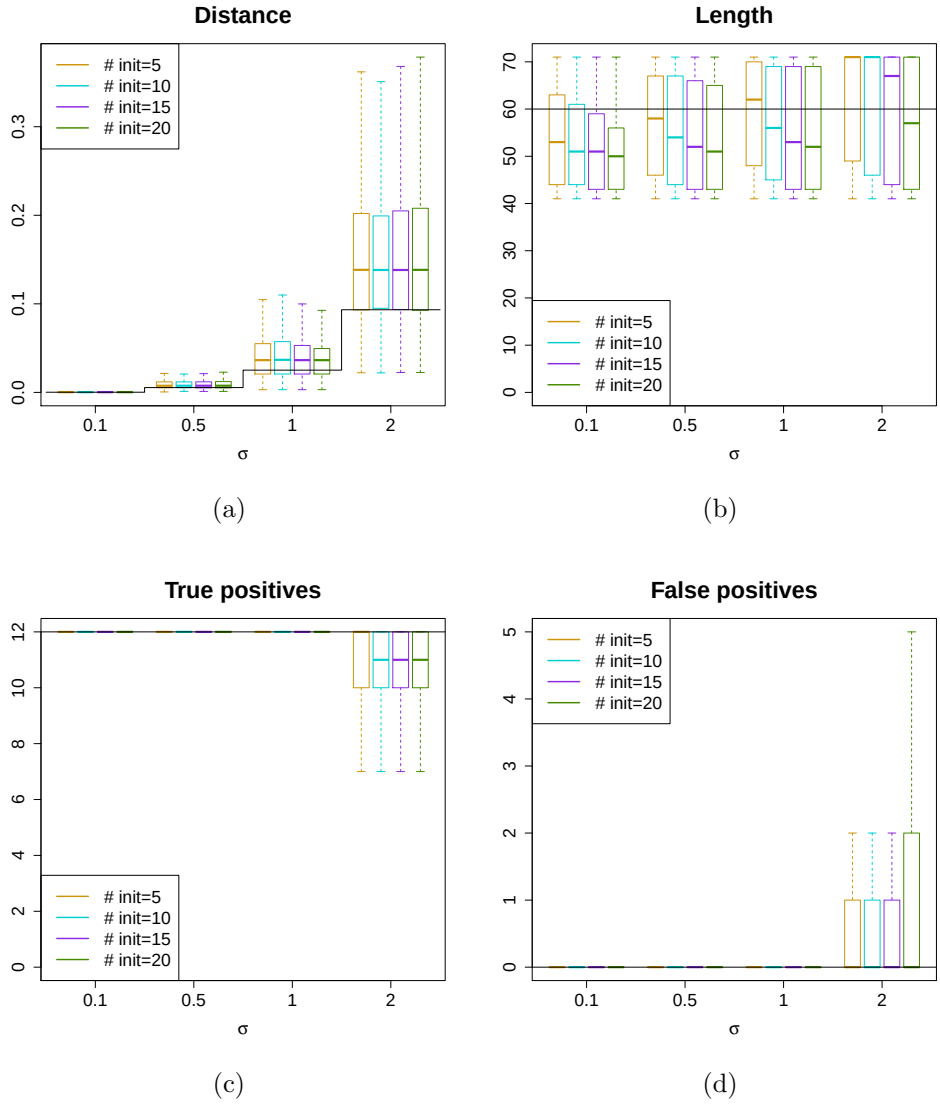


Figure S22: Summary of functional motif discovery results for the 10 datasets in scenario (1), with 5, 10, 15 or 20 initializations for each (K, c) pair. (a) Distance between true and estimated motifs; (b) Estimated length of motifs; (c) Number of true positives; (d) Number of false positives. The boxplots are obtained from 10 replications at each of the 10 different datasets, both motifs and all curve lengths (a total of 800 observations). Outliers are not plotted, for clarity of visualization.

S4.2 Comparison with time series motif discovery

Here we report additional information related to the comparison of our probKMA-based functional motif discovery to time series motif discovery (Matrix Profile), discussed in Subsection 4.3.

The definition of a motif in time series is different from the one we employ for functional data. While our functional motifs recur across a set of curves, and possibly within individual curves in the set, time series motifs are defined as recurring within a single time series, usually starting from pairs of highly similar subseries. In particular, given a time series, a motif length c, and a radius R, Lin et al. (2002) define the most significant motif 1-motif as the subsequence of length c that has the highest count of matches, i.e. of pieces in the time series with a distance less than R. Mueen et al. (2009) defines the most basic variant of 1-motif pair as the most similar pair of pieces of length c in a time series. Yeh et al. (2016, 2018) propose an algorithm, called Matrix Profile, to retrieve the nearest neighbor of every subsequence of length c. This information is used to select the top motif pairs in the time series. For each motif pair, all neighbors within distance R (i.e. all motif pair matches) can then be retrieved. It must be noted that available time series motif discovery tools are not statistical in nature and they do not estimate the level of noise of each motif. The user needs to provide one, and usually only one, motif radius R as input to the discovery procedure. On the contrary, probKMA-based discovery learns an appropriate radius for each motif from the data.

We compare our probKMA-based functional motif discovery to Matrix Profile, as implemented by the algorithm SCRIMP that is available online at `http://www.cs.ucr.edu/~eamonn/MatrixProfile.html`. We employ a slightly modified version of this code, in which the radius R is provided as input by the user (the original code fixes $R = 1$) and the tool provides as output a maximum of 100 neighbors for each of the top 3 motif pairs found (the original code only provides a maximum of 10 neighbors). The distance employed is the z-normalized Euclidean distance, which is defined as the Euclidean distance between standardized subsequence and corresponds to a correlation distance between subsequences: $d(T, Q) = \sqrt{2c\,(1 - \mathrm{cor}(T, Q))}$, with c the length of T and Q.

We consider the two simulation scenarios introduced in Subsection 4.2, focusing on two specifications: the simple case of short curves and low noise level ($\ell = 200$ and $\sigma = 0.1$), and the complex case of long curves and high noise level ($\ell = 500$ and $\sigma = 2$). For probKMA-based discovery, we use the same parameters as in Subsection 4.2 ($K = 2, 3$, minimum motif lengths $c_{min}c = 40, 50, 60$, and 20 random initializations for each (K, c_{min})). We run Matrix Profile on one time series obtained by concatenating the 20 curves one after the other and using different choices of radius (from $R = 1$ to $R = 150$). Since the tool requires also the motif length as input, we use the true motif length $c = 60$ (a newer implementation of Matrix Profile, introduced in Linardi et al., 2018, can find all motif pairs in a given range of lengths).

Table S1 reports results for simulation scenario (1), discussed in Subsection 4.3. Table S2 shows the results for an additional simulation in scenario (1), using motifs in Fig. S17(s)-(t), while Table S3 reports results of the comparison between probKMA-based functional motif discovery and Matrix Profile, for scenario (2). The results are similar to the ones in Table S1: Matrix Profile works very well in the simple case ($\ell = 200$, $\sigma = 0.1$), but in the complex case ($\ell = 500$, $\sigma = 2$) it fails to recognize one motif, while at the same time it includes many false positives; probKMA-based functional motif discovery achieve a good performance in both cases. In addition, we observe how the choice of the radius R is of utmost importance in Matrix Profile: if the radius is too small, not all occurrences are found, while if it is too large many false positives might be included. Importantly, the optimal value for the radius depends not only on motif length and on the level of noise, but also on the shapes of the motifs. Indeed, data employed in Tables S1 and S2 have exactly the same motif length and noise level, but the optimal values of radius in the simple case ($\ell = 200$, $\sigma = 0.1$) are $R = 30$ and $R = 90$, respectively.

Table S1: Comparison of probKMA-based functional motif discovery and Matrix Profile on simulation scenario (1) (TP: true positives; FP: false positives). For probKMA, we report median results (and median absolute deviations) across 10 repeated simulations.

			probKMA	Matrix Profile										
	Radius		—	1	10	20	30	40	50	70	90	110	130	150
$\ell = 200$	Motif 1	TP	**12 (0)**	2	6	8	**12**	12	12	12	12	12	12	12
		FP	**0 (0)**	0	0	0	**0**	0	0	0	0	0	0	0
$\sigma = 0.1$	Motif 2	TP	**12 (0)**	2	7	10	**12**	12	12	12	12	12	12	12
		FP	**0 (0)**	0	0	0	**0**	0	0	0	0	0	0	0
$\ell = 500$	Motif 1	TP	**11 (0.7)**	0	0	0	0	0	0	0	1	2	2	2
		FP	**2 (1.5)**	2	5	8	8	9	10	13	17	19	24	27
$\sigma = 2$	Motif 2	TP	**12 (0)**	2	2	2	12	12	12	12	12	10	12	12
		FP	**1 (1.5)**	0	1	2	8	16	23	34	51	71	82	88

Table S2: Comparison of probKMA-based functional motif discovery and Matrix Profile on an additional simulation in scenario (1) using motifs in Fig. S17(s)-(t) (TP: true positives; FP: false positives). For probKMA, we report median results (and median absolute deviations) across 10 repeated simulations.

			probKMA	Matrix Profile										
	Radius		—	1	10	20	30	40	50	70	90	110	130	150
$\ell = 200$	Motif 1	TP	**12 (0)**	0	4	6	6	6	6	6	**12**	12	12	12
		FP	**0 (0)**	2	0	0	0	0	0	0	**0**	0	0	0
$\sigma = 0.1$	Motif 2	TP	**12 (0)**	2	9	12	12	12	12	12	**12**	12	12	12
		FP	**0 (0)**	0	0	0	0	0	0	0	**0**	0	0	0
$\ell = 500$	Motif 1	TP	**11.5 (0.7)**	0	0	0	0	0	0	1	1	1	0	1
		FP	**0 (0)**	2	2	4	4	5	8	10	17	22	27	35
$\sigma = 2$	Motif 2	TP	**10 (1.5)**	2	4	7	8	8	12	12	7	7	12	12
		FP	**1 (0)**	0	0	1	2	2	5	13	12	13	29	34

Table S3: Comparison of probKMA-based functional motif discovery and Matrix Profile on simulation scenario (2) (TP: true positives; FP: false positives). For probKMA, we report median results (and median absolute deviations) across 10 repeated simulations.

			probKMA	Matrix Profile										
		Radius	—	1	10	20	30	40	50	70	90	110	130	150
$\ell = 200$	Motif 1	TP	**12 (0)**	2	6	8	**12**	12	12	12	12	12	12	12
		FP	**0 (0)**	0	0	0	**0**	0	0	0	0	0	0	0
$\sigma = 0.1$	Motif 2	TP	**12 (0)**	2	7	11	**12**	12	12	12	12	12	12	12
		FP	**1 (0)**	0	0	0	**0**	0	0	0	0	0	0	0
$\ell = 500$	Motif 1	TP	**12 (0)**	0	0	0	0	0	0	0	0	2	3	6
		FP	**0 (0)**	2	2	2	4	6	8	16	27	37	53	68
$\sigma = 2$	Motif 2	TP	**12 (0)**	2	4	8	11	11	11	12	12	12	12	12
		FP	**2 (0)**	0	0	0	1	2	7	13	18	25	29	36

S4.3 Comparison with non-sparse and sparse functional clustering methods

We consider 2 clusters and generate 9 curves for each cluster, in the following four different scenarios (depicted in Fig. S23): (a) curves in the two clusters are aligned and they differ on the entire domain; (b) curves in the two clusters are misaligned and they differ on the entire domain; (c) curves in the two clusters differ on a portion of the domain and this portion is aligned; (d) curves in the two clusters differ on a portion of the domain and this portion is misaligned. These four scenarios can be seen as special cases of the more general functional motif discovery problem, in which each curve contains exactly one motif and (a) curves are themselves the entire aligned motifs; (b) curves are themselves the entire misaligned motifs; (c) curves contain aligned motifs; (d) curves contain misaligned motifs.

In each scenario, we run the standard functional K-means (Tarpey and Kinateder, 2003), the K-means with (global) alignment of Sangalli et al. (2010), the sparse clustering technique of Floriello and Vitelli (2017), and probKMA with Euclidean distance and $K = 2$. For the sparse clustering method, we also take the sparsity parameter (i.e. the minimum length of the unselected part of the domain) as known, setting it to the curve length minus the motif length. We set the motif length parameter in probKMA in the same way. In K-means with (global) alignment and probKMA, we consider only shift alignments. We then evaluate clustering results by means of a classification error rate (1 minus the Rand index; Rand, 1971) that is equal to 0 if every curve is correctly classified and (since $K = 2$) is equal to 0.5 if the classification is as good as random. Since probKMA produces a probabilistic clustering, we compute the classification error rate after assigning each curve to the cluster with highest membership probability. Results are shown in Table S4.

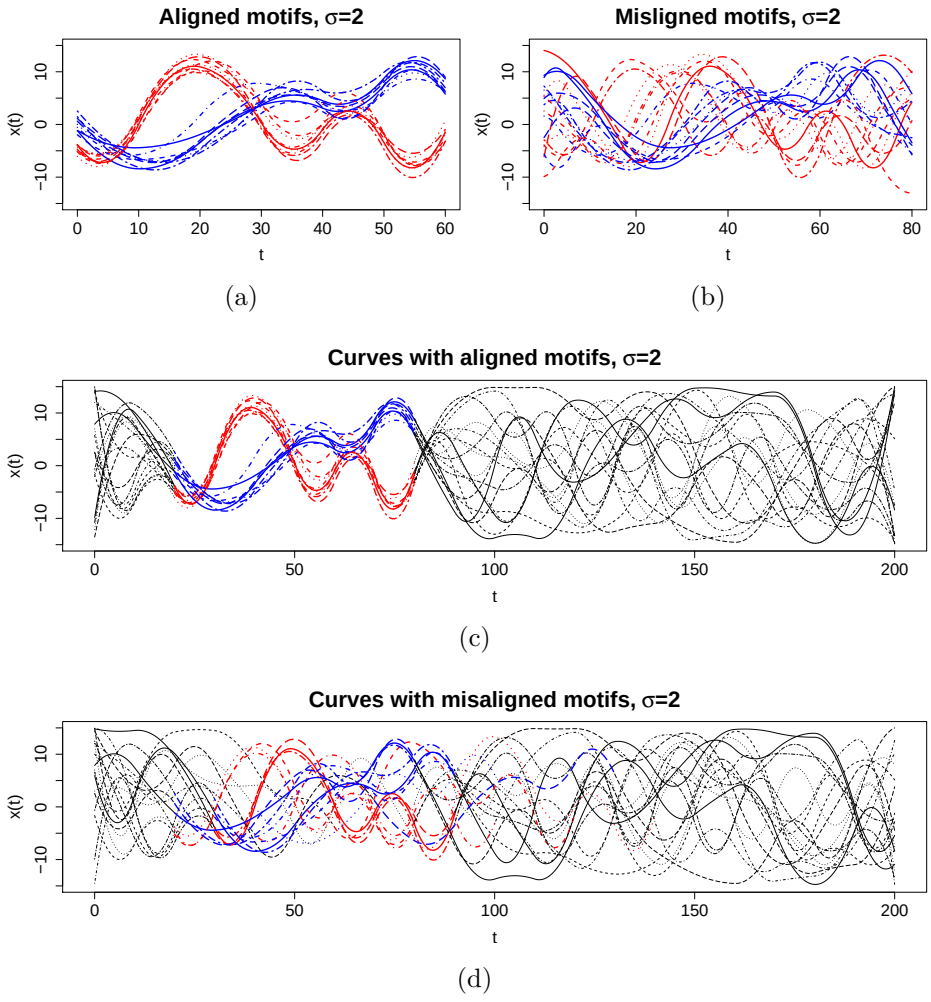
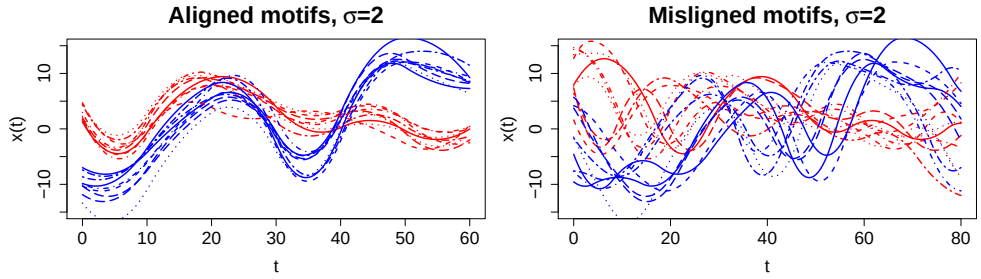


Figure S23: Simulated data for the comparison of functional clustering methods, $\sigma = 2$. (a) Aligned motifs; (b) Misaligned motifs; (c) Curves with aligned motifs; (d) Curves with misaligned motifs. The motifs are shown in red and blue and the reminder of the curves in black.

Table S4: Comparison of probKMA with functional clustering methods in the four simulation scenarios of Fig. S23. We report means (and standard deviations) of classification error rates across 10 repetitions.

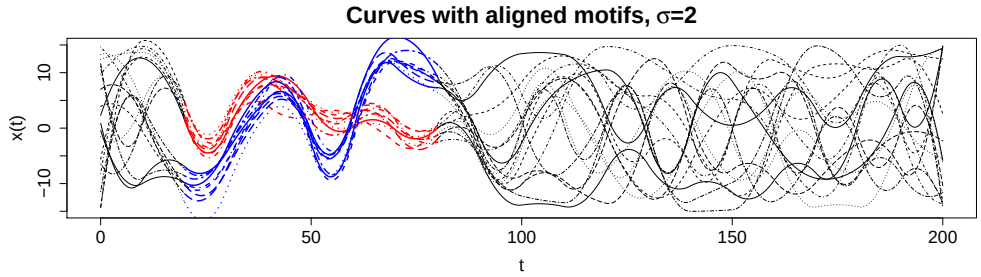
	Scenario	K-means	K-means with (global) alignment	sparse clustering	probKMA
$\sigma = 0.1$	(a)	**0 (0)**	**0 (0)**	**0 (0)**	**0 (0)**
	(b)	0.26 (0.13)	0.12 (0.19)	0.08 (0.18)	**0 (0)**
	(c)	0.29 (0.22)	0.44 (0.08)	0.05 (0.17)	**0.04 (0.07)**
	(d)	0.49 (0.05)	0.49 (0.06)	0.52 (0)	**0.01 (0.04)**
$\sigma = 2$	(a)	**0 (0)**	**0 (0)**	**0 (0)**	**0 (0)**
	(b)	0.26 (0.18)	0.16 (0.21)	0.42 (0)	**0 (0)**
	(c)	0.28 (0.23)	0.38 (0.17)	0.11 (0.22)	**0.04 (0.10)**
	(d)	0.44 (0.07)	0.49 (0.05)	0.53 (0.01)	**0.06 (0.08)**

Fig. S24 and Table S5 show an additional comparison between standard functional K-means, K-means with (global) alignment, sparse clustering and probKMA, in four scenarios: (a) curves in the two clusters are aligned and they differ on the entire domain; (b) curves in the two clusters are misaligned and they differ on the entire domain; (c) curves in the two clusters differ on a portion of the domain and this portion is aligned; (d) curves in the two clusters differ on a portion of the domain and this portion is misaligned.

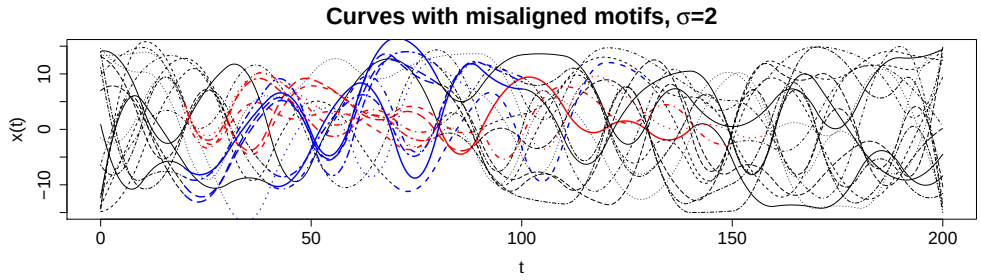


(a)

(b)



(c)



(d)

Figure S24: Data for an additional comparison of functional clustering methods, $\sigma = 2$. (a) Aligned motifs; (b) Misaligned motifs; (c) Curves with aligned motifs within them; (d) Curves with misaligned motifs within them. When the curves are broader than the motifs defining the two clusters, the motifs are shown in red and blue and the reminder of the curves in black.

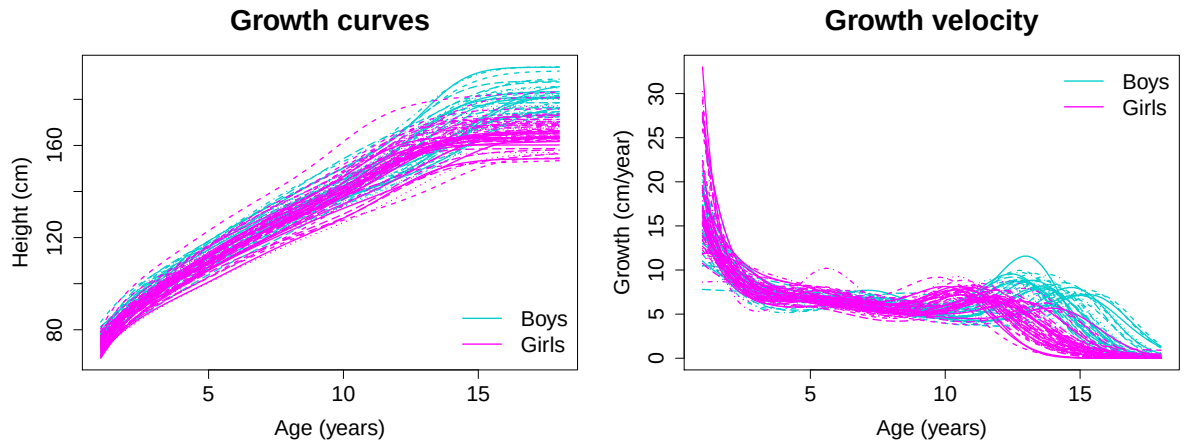
Table S5: Additional comparison of probKMA with non-sparse and sparse functional clustering methods in the four simulation scenarios of Fig. S24. We report means (and standard deviations) of classification error rates across 10 repetitions.

	Scenario	K-means	K-means with (global) alignment	sparse clustering	probKMA
$\sigma = 0.1$	(a)	**0 (0)**	**0 (0)**	**0 (0)**	**0 (0)**
	(b)	0.11 (0)	0.08 (0.15)	0.11 (0)	**0 (0)**
	(c)	0.28 (0.14)	0.40 (0.19)	0.14 (0.22)	**0.08 (0.10)**
	(d)	0.48 (0.07)	0.48 (0.07)	0.50 (0)	**0.13 (0.11)**
$\sigma = 2$	(a)	**0 (0)**	**0 (0)**	**0 (0)**	**0 (0)**
	(b)	0.11 (0)	0.12 (0.21)	0.11 (0)	**0 (0)**
	(c)	0.43 (0.10)	0.35 (0.15)	0.19 (0.24)	**0.12 (0.10)**
	(d)	0.46 (0.08)	0.42 (0.09)	0.46 (0.05)	**0.09 (0.10)**

S5 Real data applications: additional results

S5.1 Global and local clustering of Berkeley growth curves

The Berkeley Growth Study (provided by the R package `fda`) consists of the heights of 39 boys and 54 girls from age 1 to 18. We estimate height curves and their derivatives (growth velocity curves) using monotone B-spline smoothing with order 6, knots at observed ages, roughness penalty on third derivative, and $\lambda = 1/\sqrt{10}$, as suggested in the Subsection 5.2.5 of Ramsay et al. (2009). After smoothing, each curve is evaluated at 101 equidistant times between 1 and 18 years, in order to obtain 100 sub-intervals (the age difference between consecutive time points is exactly 0.17 years). Fig. S25 shows the resulting smoothed height and growth velocity curves.



(a) (b)

Figure S25: Smoothed Berkeley Growth Study curves, color-coded according to the children sex (39 boys in cyan and 54 girls in magenta). (a) Height curves; (b) Growth velocity curves (height curve derivatives).

First, we perform a global probabilistic K-means running probKMA with $K = 2$ and L^2-like pseudo-distance $d_1(\cdot, \cdot)$ between the entire curves (no alignment permitted). Note that this is equivalent to employing the L^2-like distance $d_0(\cdot, \cdot)$ on the growth velocity curves. We run probKMA 10 times, using different random initializations. Notably, all 10 initializations produced the same results (the differences are in the order of 10^{-7} for the functional J_m,

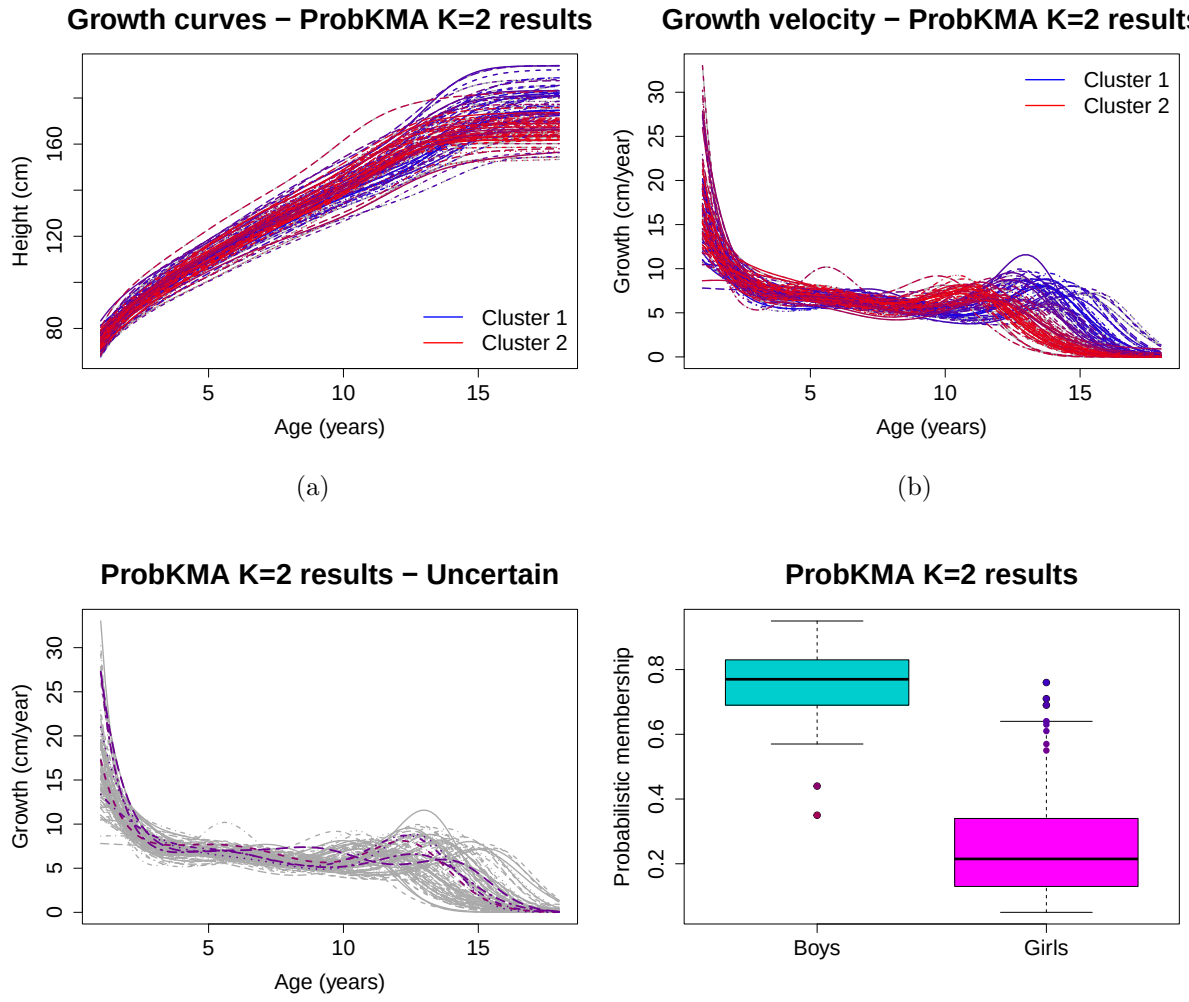
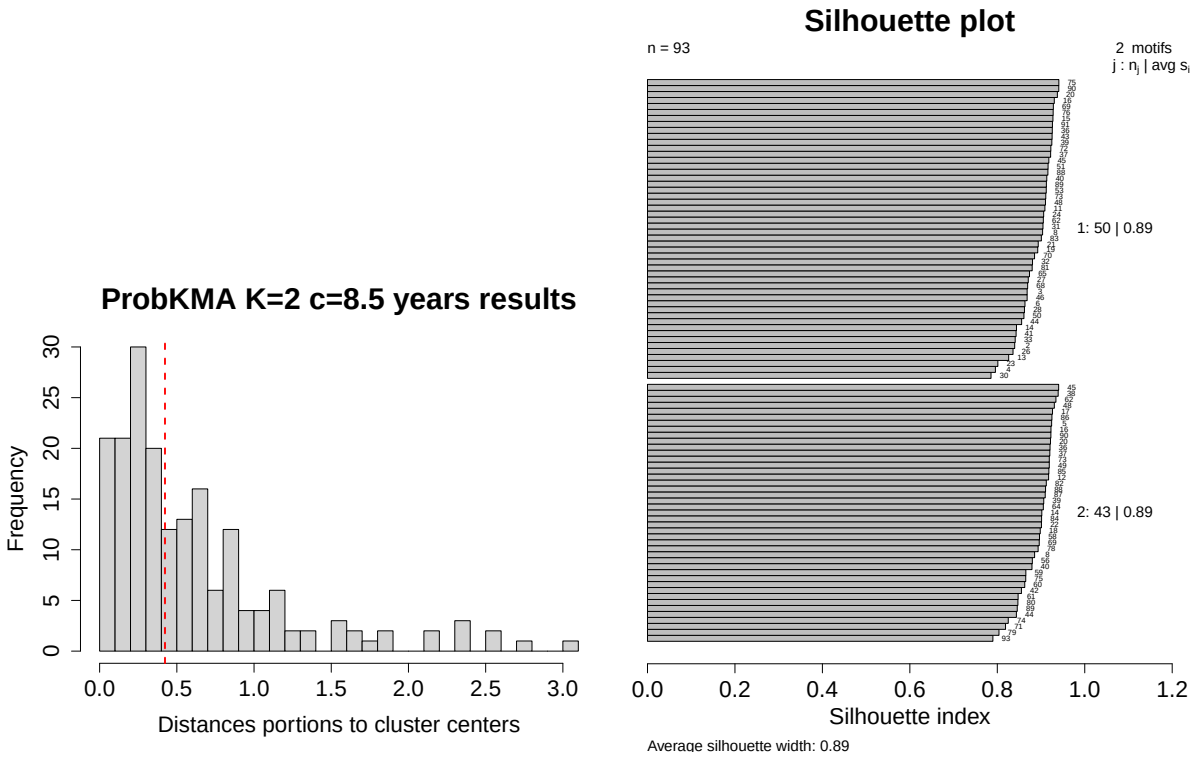


Figure S26: Results of global probabilistic K-means (with $K = 2$) for the Berkeley Growth curves, color-coded based on the membership to Cluster 1 (from red when it is 0 to blue when it is 1). (a) Growth curves; (b) Growth velocities; (c) Growth velocities for curves whose membership is uncertain (probabilistic membership between 0.4 and 0.6); (d) Probabilistic memberships for boys and girls. Dots show misclassified children.

and maximum 10^{-4} for the probabilistic memberships). Results are shown in Fig. S26: the clustering is based on the timing of the main pubertal growth spurt, that generally happens in advance for girls. Children with uncertain memberships (Fig. S26(c)) have intermediate pubertal growth spurt timing. Assigning each curve to the cluster with highest membership

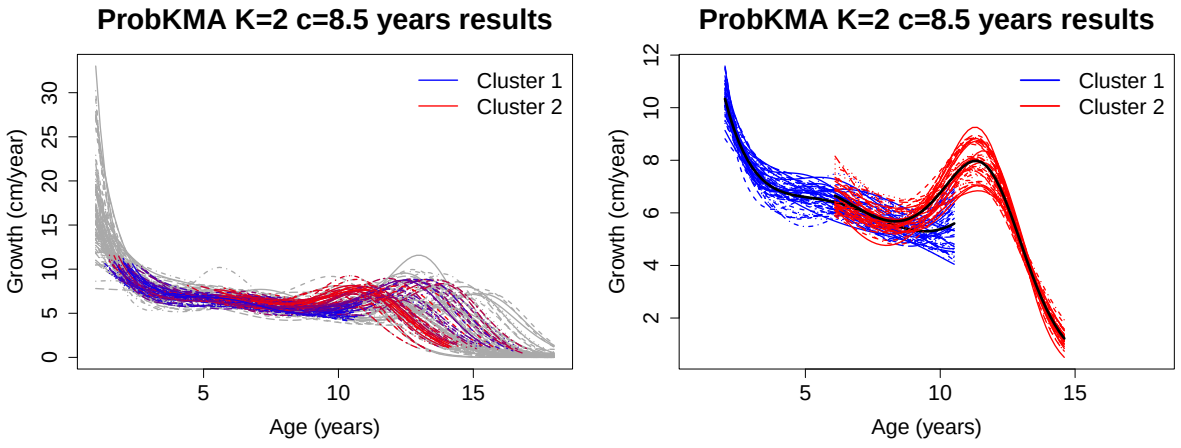
probability, we obtain clusters that differ on the main pubertal growth spurt timing and roughly corresponds to boys and girls (boys grow later), with 11 misclassified children (2 boys and 9 girls, classification error rate 0.21, Fig. S26(d)). Notably, classic functional K-means with $K = 2$ and Euclidean distance between height curve derivatives (i.e., Euclidean distance between growth velocity curves) produces exactly the same clusters. However, our probabilistic approach permits, in addition, to visualize curves whose membership is uncertain and to check the probabilistic memberships of misclassified children.



(a) (b)

Figure S27: Cluster assignments in local probabilistic clustering with portions of length 8.5 years, based on the median distance between curve portions and cluster centers. (a) Histogram of distances between curve portions and cluster centers. The red vertical bar indicates the median distance used for probabilistic membership dicotomization; (b) Silhouette plot of the resulting clustering.

To fully exploit probKMA and gain additional insights, we perform a second analysis



(a) (b)

Figure S28: Local clustering results for Berkeley curves. (a) Growth velocity curves, with clustered portions of $c = 8.5$ years color-coded based on the probabilistic membership to Cluster 1 (from red when it is 0 to blue when it is 1); (b) Cluster centers (black) with aligned clustered portions.

in which we cluster curves locally, using again the L^2-like pseudo-distance $d_1(\cdot, \cdot)$ between growth velocity curves, looking for $K = 2$ clusters of fixed length $c = 51$, corresponding to 8.5 years (hence allowing for a maximum shift of 8.5 years). We run the algorithm 10 times, using different initializations, and we select the results corresponding to the minimum value for the function J_m. We perform cluster assignment dicotomizing the probabilistic membership matrix $\mathbf{P}$ based on the median of all distances $d_1(\tilde{\mathbf{x}}_i, \mathbf{v}_k)$, i.e., we set them to 1 when the distance between the portion of curve and the cluster center is lower than the median distance (see Section S2 and Fig. S27). The silhouette plot of Fig. S27(b) shows that this clustering has a good overall quality, as indicated by the high overall average silhouette index $S = 0.89$ (reported at the bottom of the plot); both clusters are good, since they both have high average silhouette index ($S_1 = 0.89$ and $S_2 = 0.89$ for the two clusters, respectively, as indicated on the right of the plot); finally, all generalized silhouette indices s_j are positive and rather large (as indicated by the gray bars in the plot), hence all portions are appropriately assigned. As a result, we obtain two clusters of 50 and 43 portions, respectively. Interestingly, 32 and 25 curves belong exclusively to Cluster 1 and

2, respectively, while 18 curves belong to both clusters (i.e., each of these 18 curves has a portion belonging to Cluster 1 and another portion, possibly overlapping the first one, belonging to Cluster 2); 18 curves do not belong to any cluster. Since the curves are aligned and the pubertal growth spurt differs between boys and girls only in its timing (boys grow later), this local clustering does not separate boys and girls. Instead, Cluster 2 captures a particular shape of the pubertal growth spurt, which is shared by several children, while Cluster 1 captures the decrease in growth velocity that is typical in children between 2 and 3 years of age.

S5.2 Local clustering of Italian Covid-19 excess mortality curves

We consider raw mortality data published by the Italian Institute of Statistics (ISTAT) on June 4^{th}, 2020[1] and process them as explained in Subsection 5.1. Population data in the Italian municipalities – employed to normalize the excess mortality curves in the different Italian regions – is available from ISTAT at January 1^{st}, 2019[2]. Fig. S29 shows the raw Covid-19 excess mortality rate curves for the 20 Italian regions, while Figs. S30-S31 report additional results related to the analysis presented in Subsection 5.1.

[1]Available at https://www.istat.it/it/files/2020/03/Dataset-decessi-comunali-giornalieri-e-tracciato-record-4giugno.zip.

[2]Can be downloaded from I.Stat website at http://dati.istat.it/Index.aspx (Popolazione e famiglie/Popolazione/Popolazione residente al 1° gennaio/Tutti i comuni/2019).

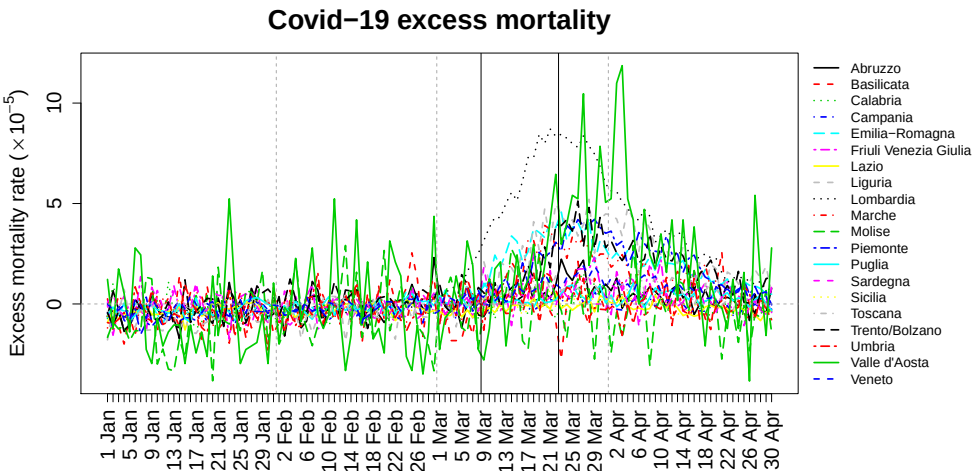
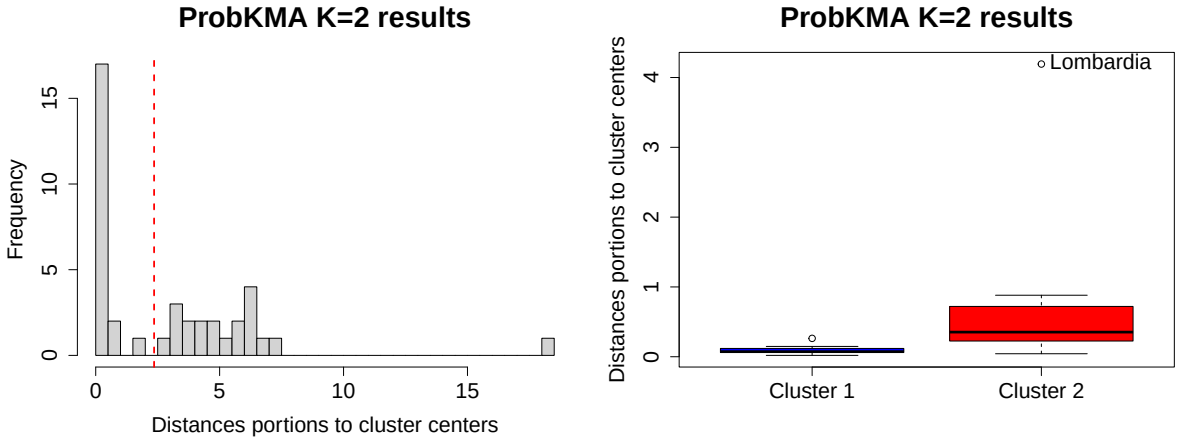
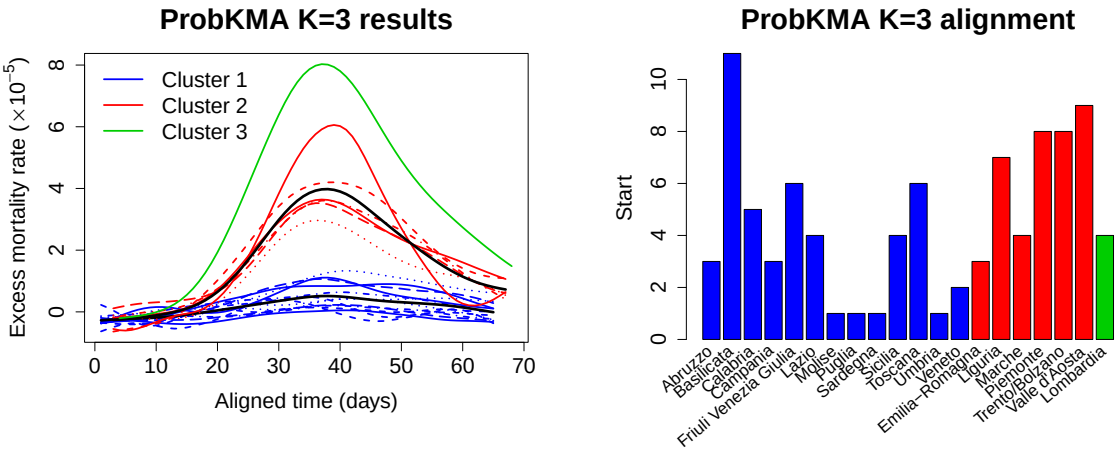


Figure S29: Raw Covid-19 excess mortality rate curves for the 20 Italian regions. Vertical black lines represent national lock down (March 9^{th}) and closure of all non-essential economic activities (March 23^{rd}).



(a) (b)

Figure S30: Distances between curve portions and cluster centers for probKMA with $K = 2$ and fixed length of $c = 65$ days. (a) Distances between all curve portions and cluster centers. The red vertical bar indicates the median distance; (b) Distances between curve portions belonging to the two clusters (assignment based on the highest probabilistic membership) and cluster centers.



(a) (b)

Figure S31: ProbKMA results with $K = 3$ and fixed length of $c = 65$ days. (a) Cluster centers (thick black curves) with aligned portions of curves; (b) Alignment between portions of curves within each cluster (start day of each portion).

S5.3 Motif discovery in mutagenesis data

We estimate high-resolution neutral mutation rates using the same pipeline as in Kuruppumullage Don et al. (2013). First, we identify neutral DNA by considering all repeats ancestral to human and macaque (AR subgenome, see e.g. Hardison et al., 2003). In particular, we consider the human reference genome *hg19* and we select all the repeats (interspersed repeats and low complexity DNA sequences) from RepeatMasker track (Smit et al., 2010) using Galaxy (Blankenberg et al., 2010; Goecks et al., 2010), excluding L1PA1-7, L1HS, AluY (primate- or human-specific elements) and Conserved Non-Exonic Elements (CNEEs, putative regulatory regions detected by Lowe and Haussler, 2012). In total, we obtain 5 407 927 AR regions, covering $\sim$ 43% of the entire genome. We then consider the 47 hot regions identified by Kuruppumullage Don et al. (2013). Since these regions are provided on the *hg18* release of the human reference genome, we use the lift-over tool (Blankenberg et al., 2010) to convert them to *hg19*. Requiring that a minimum of 90% of the nucleotides remap to the *hg19* release, we are able to retain 43 regions – corresponding to 91.5% of the initial regions. We partition these 43 regions in 1-kb windows, and we discard the ones with less than 25% AR coverage to avoid very inaccurate rate estimates. Afterwards, we extract multiple alignments corresponding to AR subgenome in each 1-kb window, using the 46-way multiZ alignment available in Galaxy (Blankenberg et al., 2011), as depicted in Fig. S32. To estimate substitution rates, we fetch pairwise alignments of human and orangutan (ponAbe2 assembly) reference genomes, masking low quality nucleotides within each block (in particular, we require an orangutan PHRED score greater than 20). Next, we identify nucleotide substitutions, i.e., the number of different nucleotides in between the aligned human and orangutan genomes. Finally, we estimate the substitution rate in each window as the number of substitutions divided by the total number of compared nucleotides in the window, using the Jukes-Cantor model (Jukes and Cantor, 1969). Although we require that at least the 25% of each considered window is covered by AR neutral subgenome, it can happen that alignments are present only in a portion of this subgenome. In this case rate estimation can be inaccurate, since a very low number of nucleotides corresponding to alignments are compared to compute it.

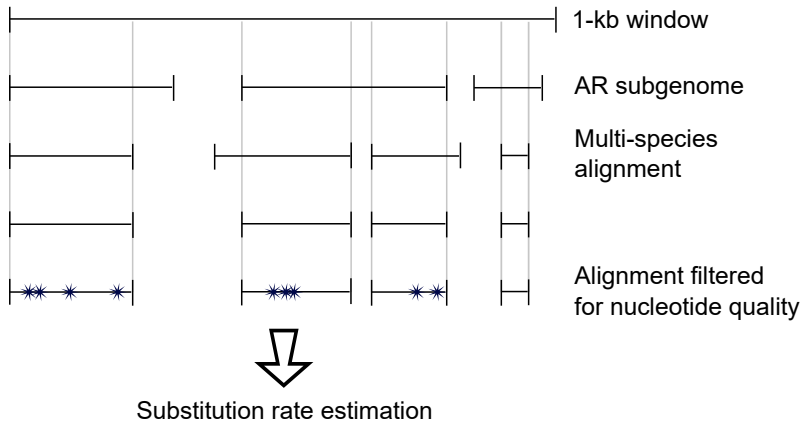


Figure S32: Schematic summary of substitution rate estimation.

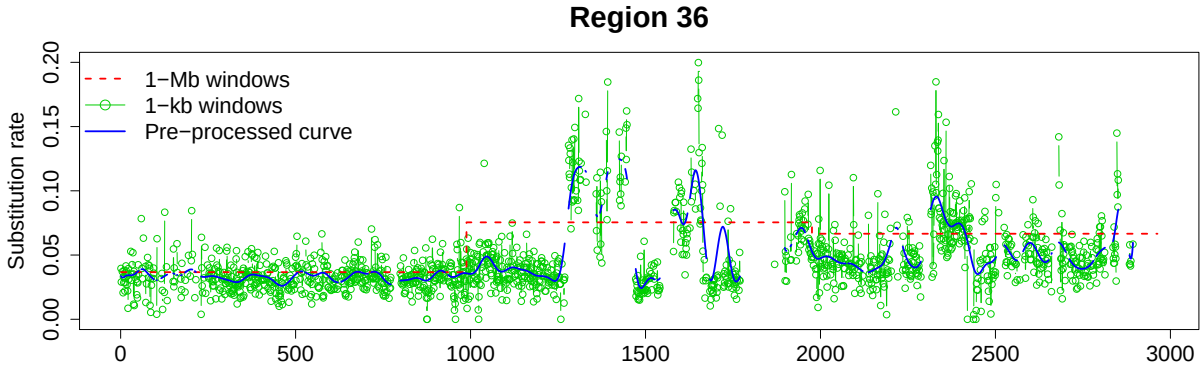


Figure S33: Example of substitution rate curve. The red dashed step function reports the rates estimated in 1-Mb windows by Kuruppumullage Don et al. (2013); the green curve with points represents the high-resolution rates estimated in 1-kb windows (only accurate values are shown); the blue curve represents the pre-processed curve, obtained after missing data imputation and local smoothing.

The resulting 43 substitution rate curves are highly noisy, and contain several missing or inaccurate values, due to the segmented nature of AR subgenome and of the multiple alignments considered to estimate rates (see, e.g., the green curve with points in Fig. S33). We assume that rates vary continuously in nearby windows along the genome, and we propose to pre-process them with stochastic regression imputation and local smoothing, filling small gaps while retaining large gaps (i.e. long stretches of missing values) for which

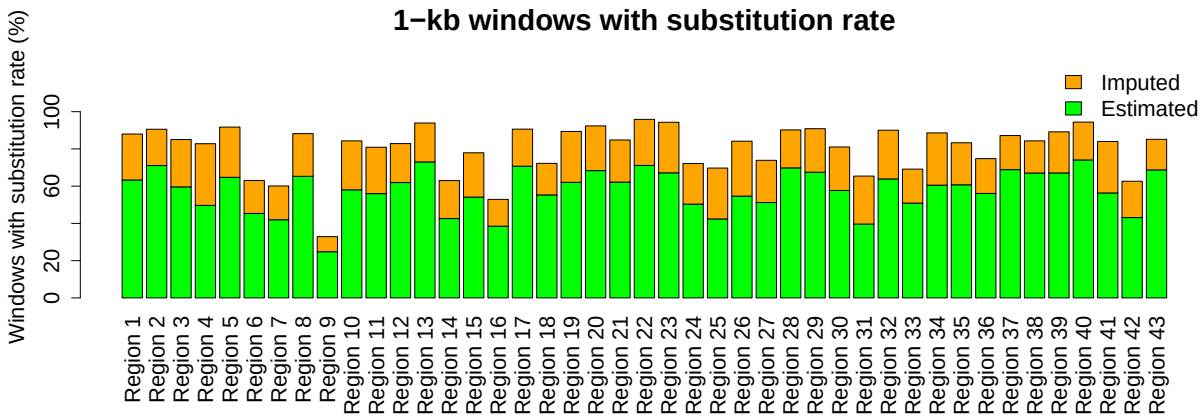


Figure S34: Quality of the substitution rate estimates in the 43 hot regions. Each bar represents the percentage of 1-kb windows with an accurate estimated rate (green) and the percentage of 1-kb windows with an imputed rate (orange).

we do not have enough information. We flag as inaccurate all rates estimated using less than 200 nucleotides, and we fill in all gaps comprising at most 5 windows using stochastic weighted regression imputation (see e.g. Enders, 2010). For each sequence of contiguous windows with missing or inaccurate rates (corresponding to a gap in the curve), we fit a weighted simple linear regression model for the rate – considering 2 neighbor windows on each side of the gaps and weighting each observation based on the reliability of the rate estimate (i.e., employing a weight proportional to the number of nucleotides used to estimate the rate) – and we impute the values according to the model predictions. Next, we add a residual noise to the imputed values, randomly sampling from the residuals of the fitted model with probabilities proportional to their weights. Notably, the vast majority of gaps in our data is quite small ($\leq$5 windows), hence with this missing-data imputation pre-processing step we are able to fill 92% of the gaps, reducing missing values to 17% of the windows (see Fig. S34). Finally, we employ local polynomials of degree 4, bandwidth 25 and Gaussian kernel on each curve to obtain a smooth functional object and compute the derivative (see the blue curve in Fig. S33).

We employ the Sobolev-like distance $\tilde{d}_{0.5}(\cdot,\cdot)$ to measure similarities between pieces of curves, requiring the length of the intersection between the domain $\tilde{D}_{i,s_{k,i}}$ of each shifted

curve and the interval $(0, c_k)$ where the cluster center is defined to be at least 80% of the interval length c_k, and at least the minimum motif length c_{min} (see Section S2). We run probKMA with $K = 2, 3, 4, 5$ and minimum lengths $c_{min} = 40, 50, 60, 70$, using 10 random initializations for each (K, c) pair. The maximum motif length c_{max} is set to 150. We note that the choice of c_{min} must be compatible with the pre-processing employed to obtain smooth curves and derivatives from discretely observed data (see also comments in the Discussion). Indeed, at very small scales we might observe "false" motifs that are artificially inserted in the curves through missing-data imputation and/or smoothing. In particular, the mutagenesis curves contain artificially-introduced "noisy straight lines" of length ≤ 5, that are employed to fill small gaps. In addition, these curves might contain more complex "false" motifs due to the local polynomials of degree 4 employed to smooth the data. However, such motifs should be quite short, since a bandwidth of 25 and a Gaussian kernel are used for local polynomials. Hence, we do not expect any "false" motif of length ≥ 40 windows – the smallest motif length we consider. Weighting fuzziness parameter is fixed to be $m = 2$, and probKMA iterations are stopped when the global Bhattacharyya distance $BC_{\max} = \max_{k=1,\dots,K} BC_k$ is $\leq 10^{-8}$. The elongation step is performed every 5 iterations, when $BC_{\max} \leq 10^{-3}$; each center is elongated up to 50% of its length in either directions, requiring that the increase of the relative objective function $J_{m,k}$ is less than 5% (i.e., that $(J_{m,k,\text{elong}} - J_{m,k})/J_{m,k} < 0.05$). The cleaning step is performed every 50 iterations, when $BC_{\max} \leq 10^{-4}$.

ProbKMA produces clusters of varying quality, and candidate motifs of different lengths (see Fig. S35(a)-(c)). Candidate motifs that belong to less than 5 curves, as well as the ones with an average cluster silhouette index lower than the 95^{th} percentile of all overall average silhouette indices, are filtered out (see Fig. S35(a)). As a result, a total of 54 motifs (out of 560 candidate motifs) are retained for post-processing phase. When computing pairwise distances between candidate motifs in the functional motif discovery post-processing (see Section S3, step 1), we require a minimum overlap of 75% of the shortest motif in each pair. Hierarchical clustering dendrogram is cut at height $2R_{all}$, where the global radius R_{all} is set equal to 20 (see Section S3, steps 3-4, and Fig. S35(d)-(e)). Default values are used

for group-specific radii R_m and motif selection in each group (see Section S3). Fig. S36 shows motifs found by probKMA-based functional motif discovery and their occurrences in the substitution rate curves, while Fig. S37 shows the genomic positions of four of the discovered motifs.

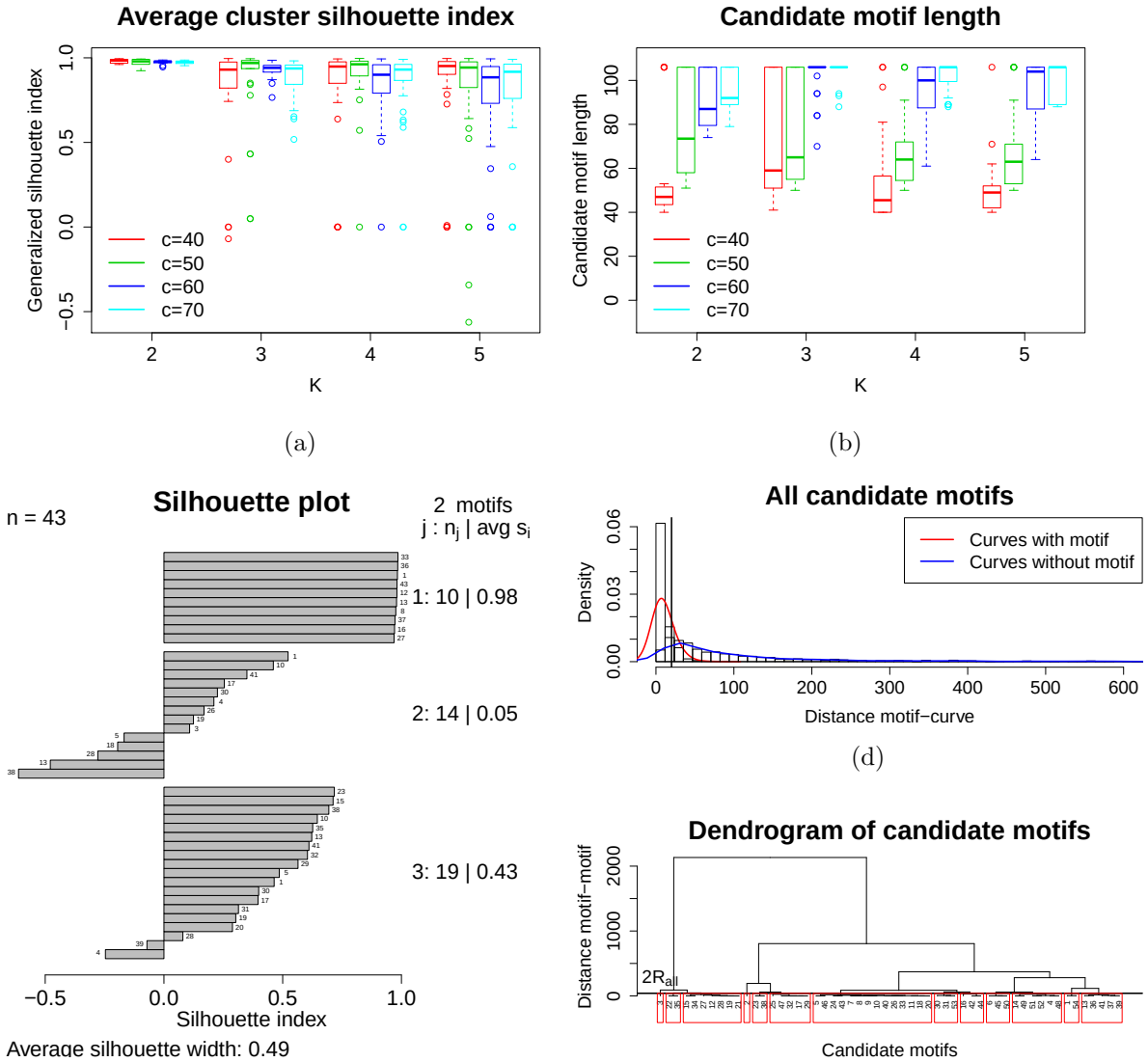


Figure S35: ProbKMA results for $K = 2, 3, 4, 5$ and $c_{min} = 40, 50, 60, 70$, and functional motif discovery post-processing results. (a) Average cluster silhouette index; (b) Candidate motif length; (c) Example of silhouette plot for $K = 3$ and $c_{min} = 50$, showing a clustering of doubtful quality (overall silhouette index $S = 0.49$), with one very good and compact candidate motif (cluster 1, with a very high average silhouette index $S_1 = 0.98$), and two bad candidate motifs (clusters 2 and 3, with very low average silhouette indices $S_2 = 0.05$ and $S_3 = 0.43$ and some bad assigned portions with negative silhouette index); (d) Selection of the global radius R_{all} used to merge similar candidate motifs in the post-processing; (e) Dendrogram for merging candidate motifs, cut at height $2R_{all}$.

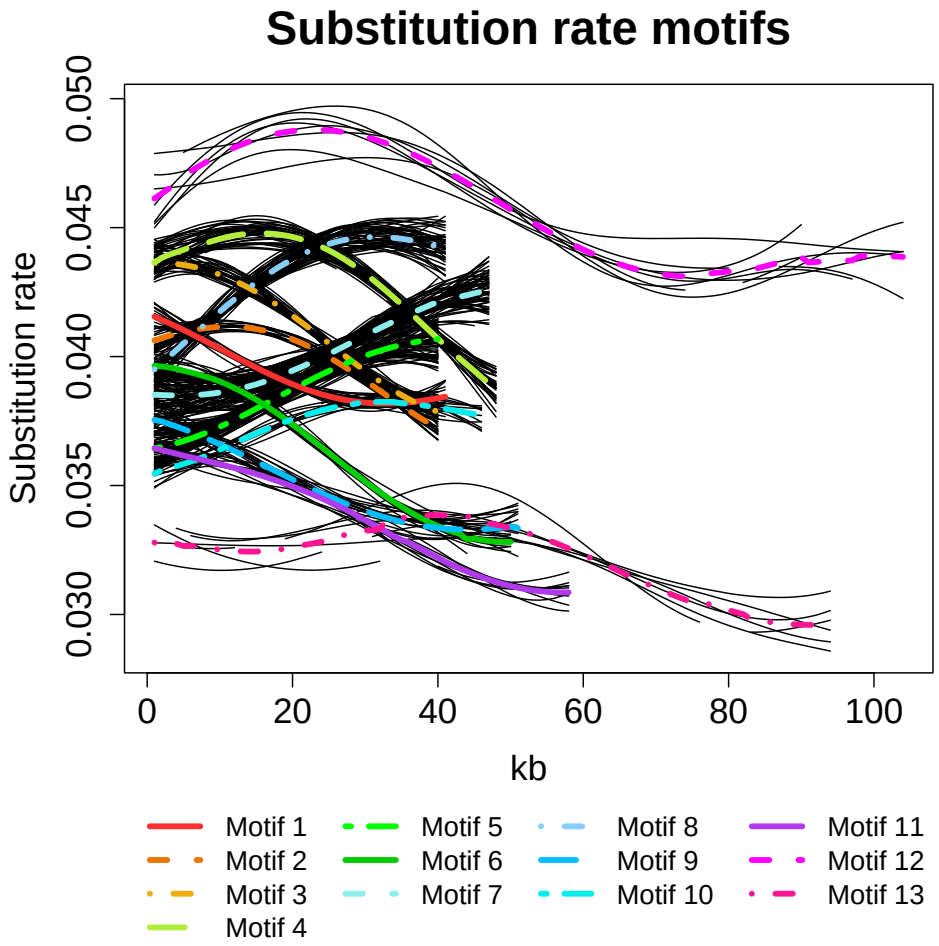
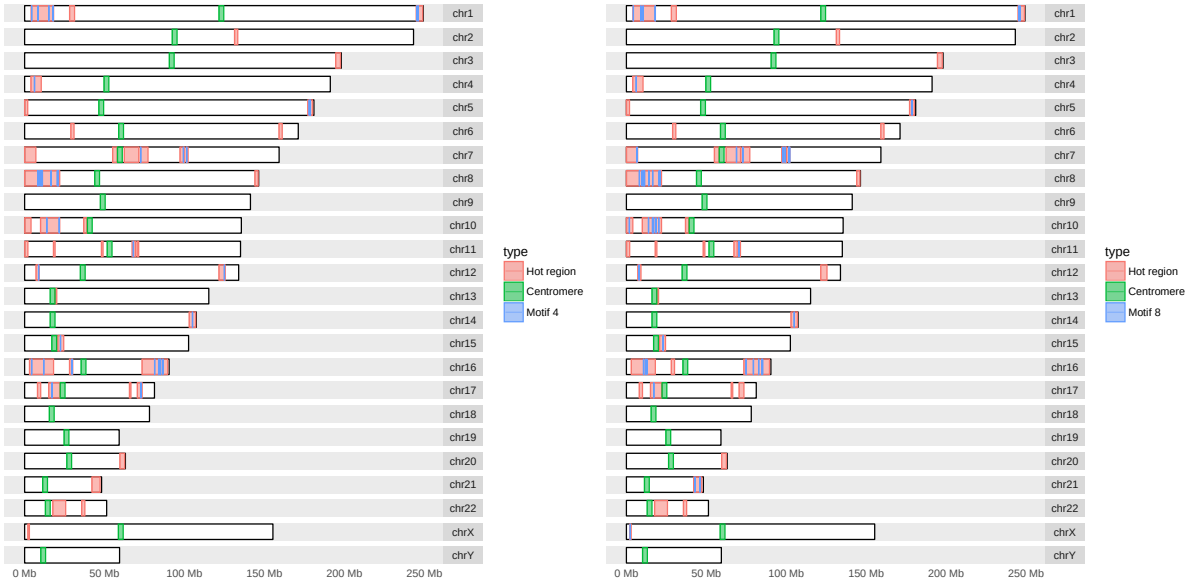
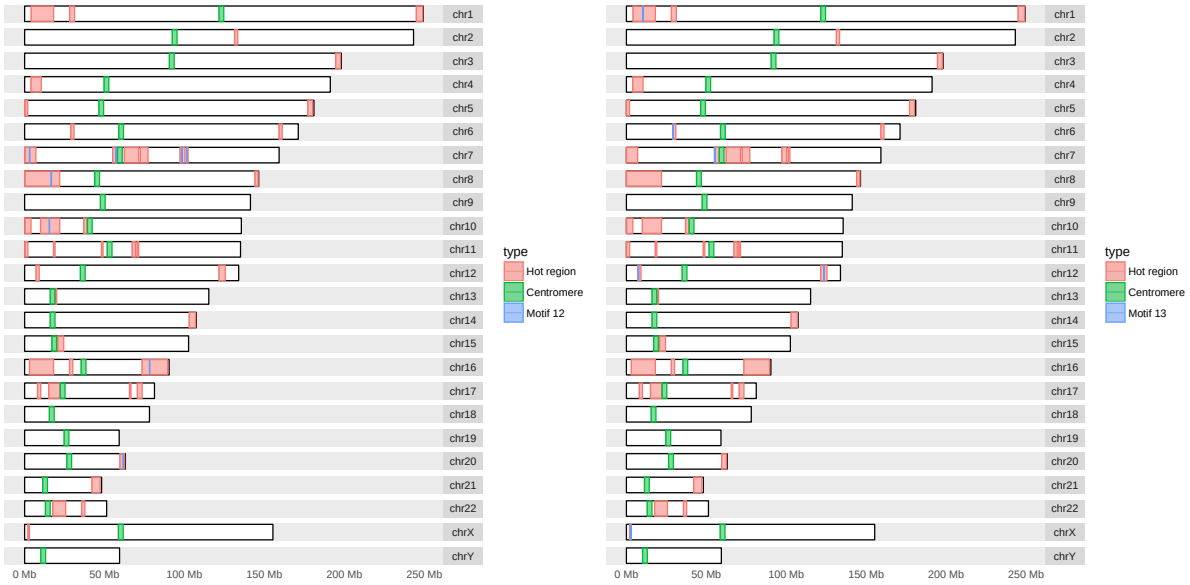


Figure S36: Results of probKMA-based functional motif discovery in substitution rate curves. The discovered motifs are plotted in their original scale (colored thick curves), together with all occurrences in the data (black solid curves).



(a) (b)



(c) (d)

Figure S37: Genomic positions of motifs occurrences (blue) in hot regions (red). (a) Motif 4; (b) Motif 8; (c) Motif 12; (d) Motif 13.

Table S6 reports the 35 genomic features considered in the investigation of the genomic landscape of the discovered motifs (see Subsection 5.2 and Fig. 5(b)).

	Type	Reference
Chromatin structure		
DNase I hypersensitive sites	Signal	ENCODE (H1-ESC)
RNA Polymerase II	Coverage	Barski et al. (2007)
CTCF	Signal	ENCODE (H1-ESC)
H2AFZ	Signal	ENCODE (H1-ESC)
Transcription regulation		
H3K27ac	Signal	ENCODE (H1-ESC)
H4K20me1	Signal	ENCODE (H1-ESC)
H3K36me3	Signal	ENCODE (H1-ESC)
H3K4me1	Signal	ENCODE (H1-ESC)
H3K4me2	Signal	ENCODE (H1-ESC)
H3K4me3	Signal	ENCODE (H1-ESC)
H3K79me2	Signal	ENCODE (H1-ESC)
H3K9ac	Signal	ENCODE (H1-ESC)
H3K9me3	Signal	ENCODE (H1-ESC)
H3K27me3	Signal	ENCODE (H1-ESC)
DNA methylation		
5-Hydroxymethylcytosine	Count	Szulwach et al. (2011)
Sperm hypomethylation	Count	Molaro et al. (2011)
Selection		
Most conserved elements	Coverage	UCSC Genome Browser
CpG islands	Coverage	UCSC Genome Browser
Exon	Coverage	UCSC Genome Browser
GC content	Percentage	Genome-wide screening
Slippage		
G-quadruplexes	Coverage	Cer et al. (2011)
A-phased repeats	Converage	Cer et al. (2011)
Direct repeats	Coverage	Cer et al. (2011)
Inverted repeats	Coverage	Cer et al. (2011)
Mirror repeats	Coverage	Cer et al. (2011)
Z DNA motifs	Coverage	Cer et al. (2011)
Mononucleotides	Coverage	Genome-wide screening

Transposition		
DNA transposons	Coverage	UCSC Genome Browser
Alu	Coverage	UCSC Genome Browser
MIR	Coverage	UCSC Genome Browser
LTR elements	Coverage	UCSC Genome Browser
Gene expression		
hESC gene expression	Weighted average	UCSC Genome Browser
Replication		
Replication origins	Count	Besnard et al. (2012)
Recombination		
Recombination hotspots	Count	Myers et al. (2008)

Table S6: List of genomic landscape features considered in Fig. 5(b).

References

Barski, A., S. Cuddapah, K. Cui, T.-Y. Roh, D. E. Schones, Z. Wang, G. Wei, I. Chepelev, and K. Zhao (2007). High-resolution profiling of histone methylations in the human genome. *Cell 129*(4), 823–837.

Besnard, E., A. Babled, L. Lapasset, O. Milhavet, H. Parrinello, C. Dantec, J.-M. Marin, and J.-M. Lemaitre (2012). Unraveling cell type–specific and reprogrammable human replication origin signatures associated with g-quadruplex consensus motifs. *Nature Structural and Molecular Biology 19*(8), 837.

Blankenberg, D., G. V. Kuster, N. Coraor, G. Ananda, R. Lazarus, M. Mangan, A. Nekrutenko, and J. Taylor (2010). Galaxy: a web-based genome analysis tool for experimentalists. *Current Protocols in Molecular Biology Chapter 19*, Unit 19.10 1–21.

Blankenberg, D., J. Taylor, A. Nekrutenko, et al. (2011). Making whole genome multiple alignments usable for biologists. *Bioinformatics 27*(17), 2426–2428.

Cer, R. Z., K. H. Bruce, U. S. Mudunuri, M. Yi, N. Volfovsky, B. T. Luke, A. Bacolla,

J. R. Collins, and R. M. Stephens (2011). Non-b db: a database of predicted non-b dna-forming motifs in mammalian genomes. *Nucleic Acids Research 39*(Database Issue), D383–D391.

Enders, C. K. (2010). *Applied missing data analysis*. Guilford Publications.

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

Goecks, J., A. Nekrutenko, J. Taylor, et al. (2010). Galaxy: a comprehensive approach for supporting accessible, reproducible, and transparent computational research in the life sciences. *Genome Biology 11*(8), R86.

Hardison, R. C., K. M. Roskin, S. Yang, M. Diekhans, W. J. Kent, R. Weber, L. Elnitski, J. Li, M. O'Connor, D. Kolbe, et al. (2003). Covariation in frequencies of substitution, deletion, transposition, and recombination during eutherian evolution. *Genome Research 13*(1), 13–26.

Jukes, T. H. and C. R. Cantor (1969). Evolution of protein molecules. In H. Munro (Ed.), *Mammalian Protein Metabolism*. New York: Academic Press.

Kuruppumullage Don, P., G. Ananda, F. Chiaromonte, and K. D. Makova (2013). Segmenting the human genome based on states of neutral genetic divergence. *Proceedings of the National Academy of Sciences 110*(36), 14699–14704.

Lin, J., E. Keogh, S. Lonardi, and P. Patel (2002). Finding motifs in time series. In *8th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, Edmonton, Alberta, Canada*.

Linardi, M., Y. Zhu, T. Palpanas, and E. Keogh (2018). Matrix profile X: VALMOD - scalable discovery of variable-length motifs in data series. In *ACM SIGMOD/PODS International Conference on Management of Data / Principles of Database Systems, Houston, Texas, USA*.

Lowe, C. B. and D. Haussler (2012). Mammalian genomes reveal novel exaptations of mobile elements for likely regulatory functions in the human genome. *PLoS One 7*(8), e43128.

Molaro, A., E. Hodges, F. Fang, Q. Song, W. R. McCombie, G. J. Hannon, and A. D. Smith (2011). Sperm methylation profiles reveal features of epigenetic inheritance and evolution in primates. *Cell 146*(6), 1029–1041.

Mueen, A., E. Keogh, Q. Zhu, S. Cash, and B. Westover (2009). Exact discovery of time series motifs. In *SIAM International Conference on Data Mining, Sparks, Nevada, USA*.

Myers, S., C. Freeman, A. Auton, P. Donnelly, and G. McVean (2008). A common sequence motif associated with recombination hot spots and genome instability in humans. *Nature Genetics 40*(9), 1124.

Ramsay, J. O., G. Hooker, and S. Graves (2009). *Functional data analysis with R and MATLAB*. Springer.

Rand, W. M. (1971). Objective criteria for the evaluation of clustering methods. *Journal of the American Statistical association 66*(336), 846–850.

Sangalli, L. M., P. Secchi, S. Vantini, and V. Vitelli (2010). K-mean alignment for curve clustering. *Computational Statistics & Data Analysis 54*(5), 1219–1233.

Smit, A., R. Hubley, and P. Green (2008-2010). RepeatMasker Open-3.0. `http://www.repeatmasker.org`.

Szulwach, K. E., X. Li, Y. Li, C.-X. Song, J. W. Han, S. Kim, S. Namburi, K. Hermetz, J. J. Kim, M. K. Rudd, et al. (2011). Integrating 5-hydroxymethylcytosine into the epigenomic landscape of human embryonic stem cells. *PLoS Genetics 7*(6), e1002154.

Tarpey, T. and K. K. Kinateder (2003). Clustering functional data. *Journal of Classification 20*(1), 093–114.

Yeh, C.-C. M., Y. Zhu, L. Ulanova, N. Begum, Y. Ding, H. A. Dau, Z. Zimmerman, D. F. Silva, A. Mueen, and E. Keogh (2018). Time series joins, motifs, discords and shapelets: a unifying view that exploits the matrix profile. *Data Mining and Knowledge Discovery 32*(1), 83–123.

Yeh, C. M., Y. Zhu, L. Ulanova, N. Begum, Y. Ding, H. A. Dau, D. F. Silva, A. Mueen, and E. Keogh (2016). Matrix profile I: all pairs similarity joins for time series: A unifying view that includes motifs, discords and shapelets. In *IEEE 16th International Conference on Data Mining, Barcelona, Spain*.