RESEARCH ARTICLE

kmlShape: An Efficient Method to Cluster Longitudinal Data (Time-Series) According to Their Shapes

Christophe Genolini[1,2]*, **René Ecochard**[3,4], **Mamoun Benghezal**[1], **Tarak Driss**[2], **Sandrine Andrieu**[1,5], **Fabien Subtil**[3,4]

1 Inserm UMR 1027, University of Toulouse III, Toulouse, France, **2** CeRSM (EA 2931), UFR STAPS, University Paris Ouest-Nanterre-La Défense, Nanterre, France, **3** Service de Biostatistique, Université Lyon 1, Villeurbanne, France, **4** CNRS, UMR5558, Equipe Biotatistique-Santé, Laboratoire de Biométrie et Biologie Evolutive, Villeurbanne, France, **5** Department of Epidemiology and Public Health, CHU Toulouse, Toulouse, France

* christophe.genolini@u-paris10.fr

Abstract

Background

Longitudinal data are data in which each variable is measured repeatedly over time. One possibility for the analysis of such data is to cluster them. The majority of clustering methods group together individual that have close trajectories at given time points. These methods group trajectories that are locally close but not necessarily those that have similar shapes. However, in several circumstances, the progress of a phenomenon may be more important than the moment at which it occurs. One would thus like to achieve a partitioning where each group gathers individuals whose trajectories have similar shapes whatever the time lag between them.

Method

In this article, we present a longitudinal data partitioning algorithm based on the shapes of the trajectories rather than on classical distances. Because this algorithm is time consuming, we propose as well two data simplification procedures that make it applicable to high dimensional datasets.

Results

In an application to Alzheimer disease, this algorithm revealed a "rapid decline" patient group that was not found by the classical methods. In another application to the feminine menstrual cycle, the algorithm showed, contrarily to the current literature, that the luteinizing hormone presents two peaks in an important proportion of women (22%).

Citation: Genolini C, Ecochard R, Benghezal M, Driss T, Andrieu S, Subtil F (2016) kmlShape: An Efficient Method to Cluster Longitudinal Data (Time-Series) According to Their Shapes. PLoS ONE 11(6): e0150738. doi:10.1371/journal.pone.0150738

Editor: Chun-Hsi Huang, University of Connecticut, UNITED STATES

Received: January 27, 2015

Accepted: February 18, 2016

Published: June 3, 2016

Data Availability Statement: All relevant data are within the paper and its Supporting Information files.

Funding: This research was funded by the Agence National de la Recherche grant IDoL: ANR-12-BSV1-0036 (CG received the funding). The funders had no role in study design, data collection and analysis, decision to publish, or preparation of the manuscript.

Competing Interests: The authors have declared that no competing interests exist.

1 Introduction

1.1 Clustering longitudinal data

Longitudinal data are data in which each variable is measured repeatedly over time. One way of analyzing this type of data is to cluster them; i.e., divide the population into homogeneous subgroups. For this, different methods were proposed among which variants of k-means [1–6] and various model-based classification methods relying on mixture models [7–11]. The pros and cons of these approaches are regularly discussed [12, 13] though there are no current recommendations on which method to prefer in a specific context.

The general idea behind partitioning is to group similar individuals within the same cluster. Different approaches to the concept of "similarity" are possible. They may be based on the concept of distance, resemblance, or likelihood. In the majority of the currently available approaches, two individuals are considered similar when they have close trajectories at each time point. This approach takes into account local similarities but not necessarily the general shapes of the trajectories. In particular, two identical trajectories but shifted in time are considered different and may be potentially assigned to distinct clusters. The immediate consequence is that the mean of the group does not inform on the shapes whereas, in a number of cases, the progress of a phenomenon may be more important than the moment at which it occurs. In such circumstances, one would prefer a partitioning that groups individuals whose trajectories have similar shapes whatever the shift in time. An example of this is shown Fig 1. With classical techniques, trajectories i_1 and i_2 (in orange) belong to the same cluster A while i_3 and i_4 (light blue) belong to another cluster B. The mean of cluster A is in red; that of cluster B is in deep blue. Using "shape-respecting clustering", i_1 and i_3 (in orange) belong to cluster A while i_2 and i_4 (light blue) belong to cluster B. The shape-respecting mean is in red for cluster A and in deep blue for cluster B.

1.2 Shape respecting tools

The problems of trajectories with similar shapes were mainly addressed in two ways: i) distances and ii) means.

Intuitively, a distance is a function that takes two individuals and returns a number. The number should have a low value when the two individuals are close and a high value when the two individuals are distant from each other. A shape-respecting distance is a distance that takes a small value when individuals have trajectories with similar shapes but a big value otherwise.

Several shape-respecting distances have been proposed in the literature. The most studied are the Fréchet distance [14] and the Dynamic Time Warping [15–18] but there are many other alternatives like HCCA [19] or EDR (Edit Distance is Real sequence) [20] or longest common subsequence [21].

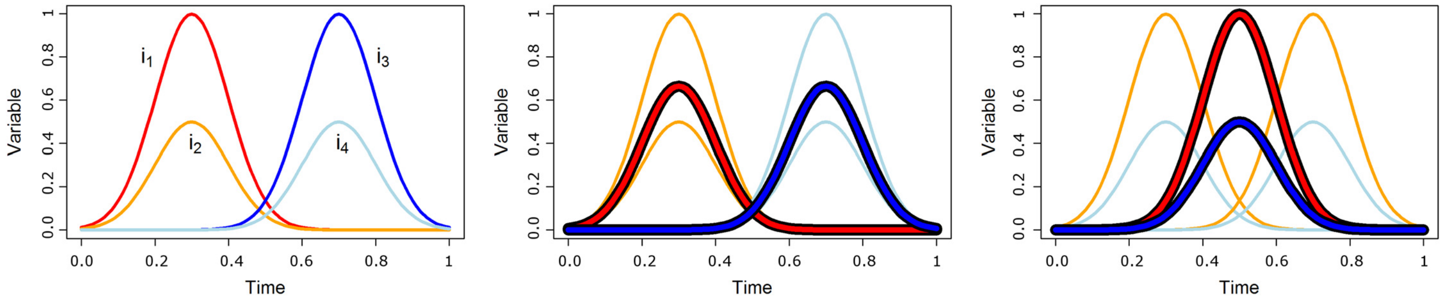


Fig 1. Cluster longitudinal data according to their shapes. (a) four trajectories. (b) With classical techniques, trajectories i_1 and i_2 (in orange) belong to the same cluster A while i_3 and i_4 (light blue) belong to another cluster B. The mean of cluster A is in red; that of cluster B is in deep blue. (c) Using "shape-respecting clustering", i_1 and i_3 (in orange) belong to cluster A while i_2 and i_4 (light blue) belong to cluster B. The shape-respecting mean is in red for cluster A and in deep blue for cluster B.

doi:10.1371/journal.pone.0150738.g001

The problem of the mean respecting the form of trajectories is more complex. Many solutions exist. Curve alignment consists in deforming the trajectories so as to align some specific points (minimums, maximums, inflexion points) [22–27]. In a second step, the deformed curves are modeled according to mixture modeling. Or a simple Euclidean means is computed on the deformed trajectory.

More recently, for a higher efficiency, a number of authors [28, 29] have chosen to partition and align jointly. Currently, they are only few articles on method performance comparisons [27, 30]; most articles tend to show that one of the most efficient methods is fdakma [31].

Unfortunately, most methods suffer from various weaknesses; mainly, they are efficient only in populations with well-separated clusters and limited shifts.

1.3 Clustering according to shape

Using k-means with a classic distance does not allow solving the similar-shape clustering problem. But using a shape distance does not allow solving it either. Indeed, the use of the shape distance will form correctly the clusters by grouping individuals whose trajectories have similar shapes, but the mean trajectory of each cluster will not necessarily be representative of the group. Thus, the following iterations are affected.

Fig 2 gives an illustration of the impact of the methods on the partitioning process. The population is shown Fig 2.a. It is a mixture of two groups of trajectories: one whose tops are high (between 0.75 and 0.85) and the other whose tops are lower (between 0.35 and 0.45). The objective of the algorithm is to identify the two groups. During the initialization phase, two individuals are randomly chosen (red and blue, Fig 2.b). The expectation phase assigns each individual to the closest cluster. By using the Euclidean distance, both individuals i_1 and i_2 are close to the red individual and will be classified in the red group whereas i_3 and i_4 will be classified in the blue group. This method leads to the partition presented Fig 2.c, then to the mean trajectories shown Fig 2.e. This partition does not find the two groups that constituted the initial population. Using a shape-respecting distance, individuals i_1 and i_3 are close to the individual in blue and will be classified in the blue group whereas i_2 and i_4 will be classified in the red group. This method leads to the partition presented Fig 2.d. Now, using a conventional way to compute the mean leads to find the mean trajectories presented Fig 2.f. The groups identified this way are correct, but the mean trajectories are not representative. The use of a shape-respecting mean leads to find the mean trajectories shown in Fig 2.g. The groups are correct and the mean trajectories are representative of the groups.

In the present article, we introduce kmlShape, a new partitioning method that clusters trajectories according to their shapes.

This method is based on a variation of the k-means algorithms in which we use a “shape-respecting distance” and a “shape-respecting mean”. Regarding the shape-respecting distance, we define a new method, the “generalized distance of Fréchet” which is a generalization of both the Fréchet distance and the Dynamic Time Warping. Regarding the shape-respecting mean, we define a new curve alignment solution. It is based on the construction of the Fréchet mean between two curves, then between n curves.

This method can be time consuming in case of large datasets. We introduce thus two methods that reduce the data size while keeping the essential information contained in the initial trajectories. The use of both data reduction and kmlShape yields a partitioning method that preserves the shapes of the trajectories and may be used with high-dimensional data.

The sections below are organized as follows: we present first the methods used to partition the trajectories according to their shapes. Next, the performances of the methods are evaluated with artificial and real data. Then we discuss the results, the quality of the partitioning on the

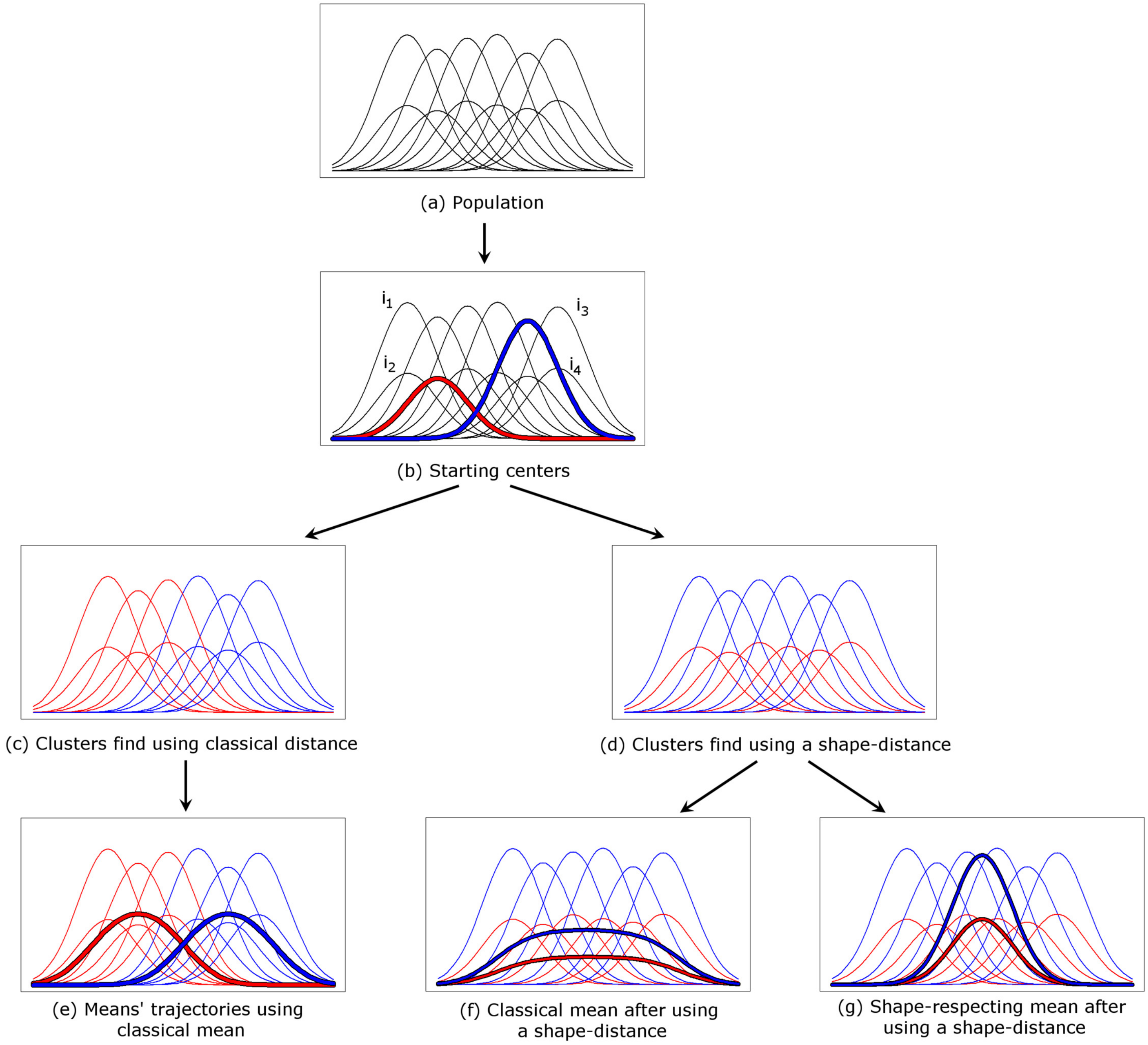


Fig 2. The impact of using the classical distance, the classical mean, the Fr´echet distance and the Fr´echet mean.

doi:10.1371/journal.pone.0150738.g002

artificial data, and the originality of the method with real data; i.e., the ability of the algorithm to reveal clusters that are undetectable with the classical methods.

2 Methods

2.1 General considerations

2.1.1 Notation. Let us consider a set S of n subjects. For each subject i, an outcome variable Y is measured t times. For the sake of simplicity, we consider that all trajectories have the same number of measurements t though trajectories with different numbers of measurements do not add complexity to the algorithm. The time of the j^{th} measurement for subject i is noted x_{ij}. The value of the j^{th} measurement for subject i is noted y_{ij}. The sequence $Y_{i.} = \left(\begin{pmatrix} x_{i1} \\ y_{i1} \end{pmatrix}, \begin{pmatrix} x_{i2} \\ y_{i2} \end{pmatrix}, \dots, \begin{pmatrix} x_{it} \\ y_{it} \end{pmatrix} \right)$ is called a trajectory.

2.1.2 k-means using shape-respecting distances. k-means is a partitioning algorithm that belongs to Classification-Expectation-Maximization (CEM) methods [32]. This algorithm was first used with classical data [33, 34] but is now widely used with longitudinal data in various fields [5, 35–40]. The principle of k-means is to alternate two steps: i) an Expectation step that calculates the distances between the individual trajectories and the mean trajectories of each cluster; then each individual is assigned to the closest cluster; ii) a Maximization step that estimates the mean trajectory of each cluster. Before alternating these two steps, an initialization phase defines the "mean trajectories of each cluster" that will be used in the first maximization step. Various initialization methods are possible, as detailed in [41–44]. Here, we use the classical method that selects randomly k inidividuals and considers them as the k first clusters' centers.

kmlShape is a new clustering algorithm that clusters trajectories according to their shape. It applies k-means within the context of a shape-respecting partitioning. As briefly reminded here, method k-means uses two tools: a distance and a mean. kmlShape requires both a distance and a mean that take the shapes into account. These tools (Fréchet distance and Fréchet means) are presented in the next section.

Overall, kmlShape is a variant of k-means using: i) the Fréchet distance to calculate the distances between individuals and cluster centers; ii) Fréchet mean to construct the centers of the clusters. The stopping condition is the stability of the algorithm: when the clusters are identical at step s and step $s - 1$, the algorithm is terminated (with a limitation of the number of iterations to avoid very long times before convergence). The pseudo code of the algorithm is given in Algorithm 1.

```
Data: Population: $n$ individuals $Y_1, \dots Y_n$
Result: Partition: $Cluster$ vector of size $n$ taking values in [1..$k$]
/* Step 0: Initialization                                                    */
$k$ individuals $C_1, C_2, \dots, C_k$ are randomly chosen in $Y_1, \dots Y_n$
$s \leftarrow 0$
$Cluster_0 \leftarrow (0, 0, \dots, 0)$ /* vector of size $n$                              */
repeat
  $s \leftarrow s + 1$
  /* Step s.1, phase expectation                                             */
  for $i$ in 1..$n$ do
    for $j$ in 1..$k$ do
      Compute $DistF_{i,j}$ (The Fréchet distance between $Y_i$ and $C_j$)
      $Clusters_S(i) \leftarrow j$ such that $DistF_{i,j}$ is smaller than $Dist_{i,j'}$ for $j' \neq j$
    end
  end
  /* Step s.2, phase maximization                                            */
  for $j$ in 1..$k$ do
    Compute $M_j$, the Fréchet mean of clusters $j$ (that is the Fréchet mean of all
      the $Y_i$ such that $Cluster_S(i) == j$)
  end
until $Cluster_S == Cluster_{S-1}$ or $s > Max\_Iteration$
```

Algorithm 1: kmlShape

2.2 Extension to the Fréchet distance

2.2.1 Fréchet distance. The Fréchet distance was introduced by Maurice Fréchet in [14]. Informally, it is often compared to a leash between two trajectories. The Fréchet distance is the minimum length of a leash that would separate a master from his dog walking at different speeds along two trajectories. In other words, each point of each trajectory is associated with the nearest point on the other trajectory. The Fréchet distance is then the longest link between the two trajectories.

Mathematically: let $d\left(\begin{pmatrix}x_1\\y_1\end{pmatrix},\begin{pmatrix}x_2\\y_2\end{pmatrix}\right)=\sqrt{(x_1-x_2)^2+(y_1-y_2))^2}$ be the Euclidian distance between points $\begin{pmatrix}x_1\\y_1\end{pmatrix}$ and $\begin{pmatrix}x_2\\y_2\end{pmatrix}$. Let P and Q be two curves from $[0, t]$ to $\mathbb{R}$. A reparameterization of the interval $[0, t]$ is a continuous function, increasing and surjective from $[0, t]$ to $[0, t]$. We denote $\mathcal{A}$ the set of all reparameterizations of $[0, t]$. Let α and β two reparameterizations in $\mathcal{A}$ and let s be a real belonging to $[0, t]$.

Intuitively, curve P can be regarded as the mobile trajectory that would travel at constant speed (e.g., two centimeter per second). So $P \circ \alpha$ is the same trajectory as P, but covered by the mobile with a variable speed, speed defined by α (e.g., α can set 1 centimeter per second as $1 \leq s \leq 3$ and then 3 centimeters per second as $3 < s \leq t$).

The distance between curves P and Q reparameterized by α and β at time s is the distance between $\begin{pmatrix}\alpha(s)\\P(\alpha(s))\end{pmatrix}$ and $\begin{pmatrix}\beta(s)\\Q(\beta(s))\end{pmatrix}$, that is $d_{\alpha,\beta,s}(P,Q)=d\left(\begin{pmatrix}\alpha(s)\\P(\alpha(s))\end{pmatrix},\begin{pmatrix}\beta(s)\\Q(\beta(s))\end{pmatrix}\right)$. The distance between P and Q reparameterized by α and β is the maximum of the distances $d_{\alpha,\beta,s}(P, Q)$ while s varies from 0 to t: $d_{\alpha,\beta}(P, Q) = Max_s(d_{\alpha,\beta,s}(P, Q))$. Then the Fréchet distance between P and Q is the smallest possible maximum between P and Q after reparameterization of P and Q: $DistFrechet(P, Q) = d_{\alpha,\beta}(P, Q)$.

The definition is the same in the discrete case with the exception that s takes values between 0 and t by intervals. Note that, contrarily to several classical distances, the calculation of Fréchet distance does not require the same number of measurements or the same time points on the two trajectories. Therefore, it can be used to cluster irregular trajectories, the use of imputation methods for longitudinal data [45–48] is not necessary.

From a computational point of view, the Fréchet distance is rather easy to determine [49] but the calculation time is longer than that required for Euclidian distance: $O(t^2)$ (the details of all the computational complexity are given in Appendix).

The generalized Fréchet distance. Fréchet has given the seminal definition within the context of two mathematical curves P and Q. Within the context of real data, there is a relative-scale issue. The variable of interest and the time variable are not measured using the same unit. This can be an important issue since a scale changes impact the Fréchet distance. Fig 3.a shows three trajectories. According to the Fréchet distance, i_1 is closer to i_2 than to i_3 (the segments that materialize the distances between the trajectories are dotted). If the scale of the X-axis is changed (Fig 3.b), i_1 will be closer to i_3 than to i_2.

This scale-change is not trivial because it impacts the partitioning. This lead to the following definition: the generalized Fréchet distance of parameter lambda between two curves P and Q is the Fréchet distance obtained after an affine transformation $A:\begin{pmatrix}x\\y\end{pmatrix}\rightarrow\begin{pmatrix}\lambda.x\\y\end{pmatrix}$; that is, $DistFrechet_{\lambda}(P,Q)=Inf_{(\alpha,\beta)\in\mathcal{A}^2}Max(P\circ\alpha\circ A, Q\circ\beta oA)$. This is what we called *the generalized Fréchet distance*. λ is the time-scale parameters.

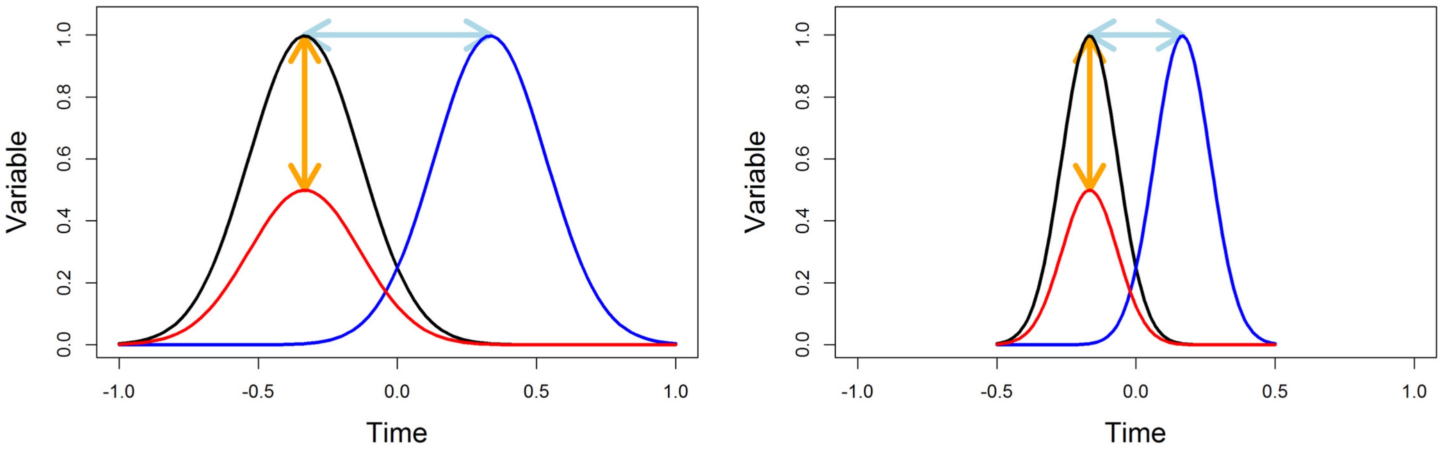


Fig 3. Scale change. (a) i_1 is closer to i_2 than to i_3 (b) i_1 is closer to i_3 than to i_2.

doi:10.1371/journal.pone.0150738.g003

One should notice that when $\lambda = 0$, the Fréchet distance matches with the Dynamic Time Warping (DTW) distance (i.e., as in DTW, horizontal shifts have no impacts. See appendix A for more details). On the opposite, when λ tends to $+\infty$, then $DistFrechet_\lambda$ tends toward the classical maximum distance.

Therefore, the generalized Fréchet distance is a generalization of shape-respecting distance (like DTW) but also of other classical distances (Maximum). Herein, for the sake of simplicity, the generalized Fréchet distance will be referred as to the Fréchet distance.

The Fréchet mean between two trajectories. As mentioned above, Classification-Expectation-Maximization algorithms require the calculation of a mean. Informally, the Fréchet mean between two trajectories is the middle of the leash that links the dog to the master when each goes along its own way.

More precisely, calculating the Fréchet distance requires the explicit calculation of the two reparameterizations α and β that minimize $DistFrechet_\lambda(P, Q)$. Using these two functions, it is obvious to define the Fréchet mean as the mean of the distances between the points of the two trajectories when these trajectories are run at speeds α and β:

$MeanFrechet_\lambda(P,Q) = \left(\begin{pmatrix} \frac{P\circ\alpha(\lambda.x(1))+Q\circ\beta(\lambda.x(1))}{2} \\ \frac{P\circ\alpha(y(1))+Q\circ\beta(y(1))}{2} \end{pmatrix}, \ldots, \begin{pmatrix} \frac{P\circ\alpha(\lambda.x(a))+Q\circ\beta(\lambda.x(a))}{2} \\ \frac{P\circ\alpha(y(a))+Q\circ\beta(y(a))}{2} \end{pmatrix}\right)$ that we will write $MeanFrechet_\lambda(P,Q) = \left(\frac{P\circ\alpha\circ A+Q\circ\beta\circ A}{2}\right)$

An example of the Fréchet mean is given Fig 4.

The Fréchet mean of the two curves P and Q weighted by coefficients p and q works on the same principle with a weighting on each curve: $MeanFrechet_\lambda(P,p,Q,q) = \frac{p.P\circ\alpha\circ A+q.Q\circ\beta\circ A}{p+q}$.

Generalization to n curves. The definition of the Fréchet mean may be extended to n curves. However, the complexity of the algorithm ($O(t^n)$) would not be realistic for the analysis of real data, even of very small size.

However, the Fréchet mean with n curves may be approximated with less complexity. The calculation of the Fréchet mean between two curves is reasonable ($O(t^2)$). In a population of n individuals, it is possible to combine pairs of individuals (with weight 1), then combine the so-obtained means (weighted by the number of individuals that generated them) until obtaining a unique mean. The calculation cost of this "step by step" mean is $O(n.t^2)$.

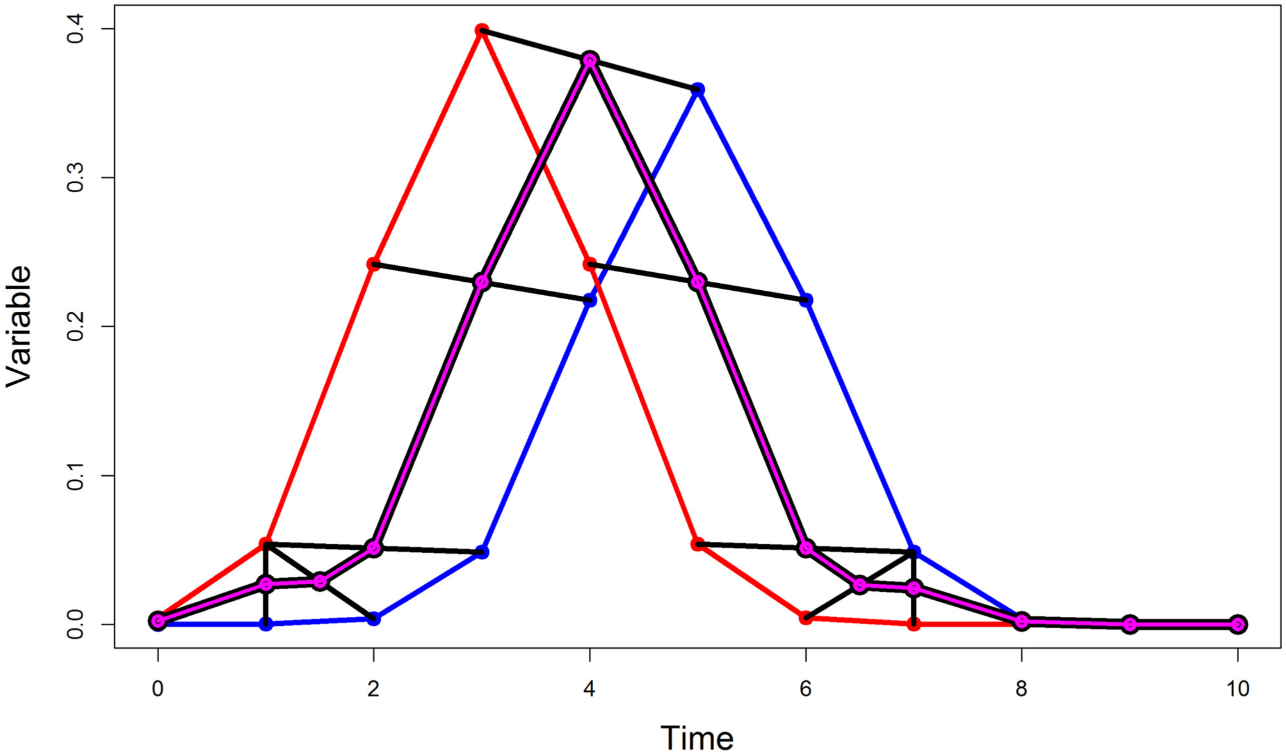


Fig 4. The Fréchet mean. Two trajectories *P* and *Q* are in red and blue, respectively. The segments linking *P* to *Q* after reparameterization are in black. The mean trajectory as defined by the middles of these segments is in violet.

doi:10.1371/journal.pone.0150738.g004

Obviously, the order in which the combinations are made has an impact on the final result. Let us mention three possible variants:

- **RandomAll**: the *n* individuals are randomly scattered on the leaves of a complete binary tree (each knots has either zero or two leaves) having depth *h* with $2^h \leq n < 2^{h+1}$. The value of each non-terminal leaf is the mean of the two children-leaves. The Fréchet mean is thus the value of the tree root. (Informally, this structure is close to that of a tennis tournament). The complexity of this method is $O(nt^2)$.
- **Hierarchical**: the combination order between individuals is fixed in a deterministic way through an ascending hierarchical classification; the closest individuals being combined first. The complexity of this method is $O(n^2\ t^2)$.
- **RandomSubset**: This method is the RandomAll method applied to a sample of randomly selected individuals. The complexity of the method is $O(n_0\ t^2)$, n_0 being the size of the random sample.

The means obtained through RandomAll and Hierarchical are very close and, in the case of simulations with artificial data, are also very close to the real mean. The choice of one of theses two methods has thus no impact on the final partitioning. On the contrary, the performance of RandomSubset is dependent on the sample size. Besides, the Hierarchical method is deterministic, which, in the case of an algorithm run several time (such as k-means) is a disadvantage because, in case of convergence toward a local maximum, an additional run of the algorithm will lead to the same maximum. Finally, its complexity is $O(n^2\ t^2)$ whereas that of randomAll is $O(nt^2)$. Thus, it is RandomAll that should be preferred.

2.3 Data size reduction

Reduction of the number of individuals. The use of the Fréchet mean approximation shifts the complexity of our first algorithm from $O(t^n)$ to $O(nt^2)$. This is an important gain, however insufficient for applying the method to large databases (thousand or tens of thousand individuals). One optimization option is to reduce the number of individuals by an identification of a small number of comparable trajectories. This suggestion of simplification is based on two facts: i) in large populations, some groups of individuals have close trajectories (because the limited number of typical trajectories); this is all the more true as the population becomes larger; ii) when two trajectories are very close, the Euclidian mean and the Fréchet mean are close (see Fig 5). It becomes then locally satisfactory to approximate the Fréchet mean through the Euclidian mean.

With these facts, in case of large populations, it is convenient to replace close groups of individuals by representatives (in the same way senators represent populations of states). In addition, the Fréchet mean may be approximated through the Euclidian mean without changing the forms of the trajectories. In the end, this reduction of the number of individuals may be obtained using a classical classification algorithm such as k-means with a Euclidian distance.

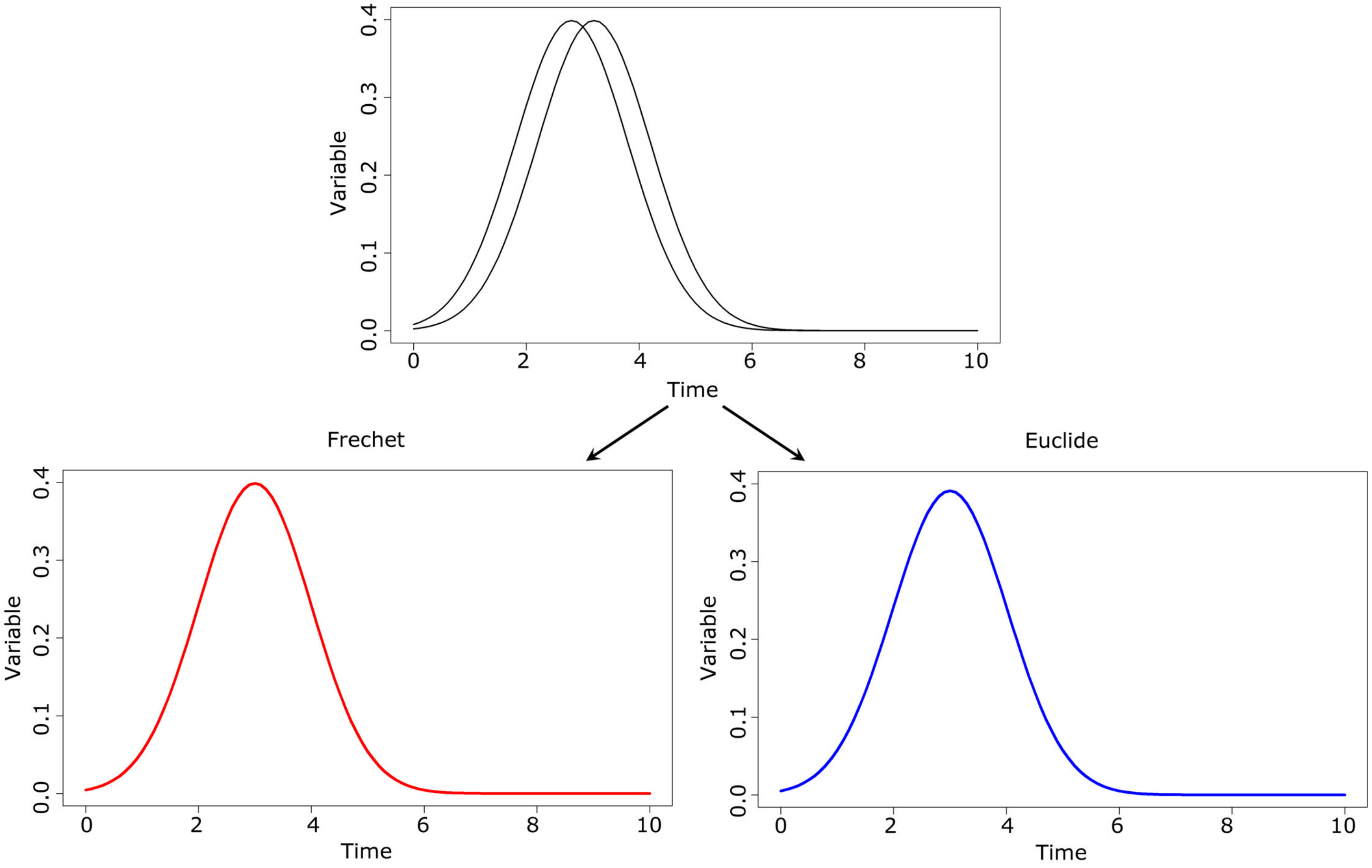


Fig 5. Comparison between Euclidian mean and the Fréchet mean in case of two close curves. The means are almost identical.

doi:10.1371/journal.pone.0150738.g005

Practically, k-means is carried out on a population of size n, with, say, $n_S = 128$ groups. This is equivalent to make the not very constraining hypothesis that there is a set of 128 representative trajectories so that each individual trajectory is close to at least one of them. The cost of this preliminary classification (we would conveniently name "election") is $O(n_S\, nt)$. Afterwards, kmlShape with weighting may be used with n_S senators stemming from the election. The cost of kmlShape is then $O(n_S\, t^2)$. The overall complexity is $O(n_S\, nt + n_S\, t^2)$.

Reduction of the number of measurements. In an orthogonal way, it is generally possible to simplify the trajectories by reducing the number of measurements made without much loss of information. These techniques are known as "Segmentation Time Series" [50, 51], "Line-simplification" [52] or "Trajectories compression" [53] In his survey, Keogh proposes three kinds of methods: the Sliding Windows, the Top-Down, and the Bottom-up. For our purpose, the Top-Down are the ones that have the best complexity. In this article, we will focus on Douglas-Peuker algorithm [54], also known as Ramers algorithm [55] or "Iterative End-Points Fits" [56].

Let us consider a trajectory Y of length t and an ϵ. The Douglas-Peuker algorithm [54] allows finding a curve Y^{DP} of length $t_{DP} \leq t$ so that the distance (projection of one point of one curve on the other curve) between Y and Y^{DP} is, in each point, less than ϵ. The Douglas-Peuker algorithm is recursive; as long as the simplified trajectory Y^{DP} is not at a distance less than epsilon from the original trajectory Y, the point of Y the farthest from Y^{DP} is added to Y^{DP}. This algorithm makes it possible to set the quality of the approximation of Y through Y^{DP}. Note that many amelioration of this algorithm exist [53].

In our present problematic, it may be more interesting to set the adequate length t_{DP} for the simplified trajectory because this length has a direct impact on the computation time. This may be obtained through a simple modification of Douglas-Peuker algorithm. Instead of considering a calculation-stopping condition that depends on the distance between Y and Y^{DP}, we may choose to set the maximum number of points for Y^{DP}: as long as the simplified trajectory Y^{DP} has less than t_{DP} points, the point of Y the farthest from Y^{DP} is added to Y^{DP}. With n_S individuals, the complexity of this algorithm is $O(t_{DP}^2 n_S t)$. Fig 6 shows the two types of simplification with 5 and 15 points, respectively.

Note that the modification of the stopping condition induces that the error is no longer explicitly controlled. In some specific cases, this might lead to a simplified trajectory that is no longer close to the initial trajectory (for example, the trajectory $\sin(t)$ with t in $[0, 3\pi]$ approximated with only 3 points). To inform the user on the size of the error, the modified Douglas-

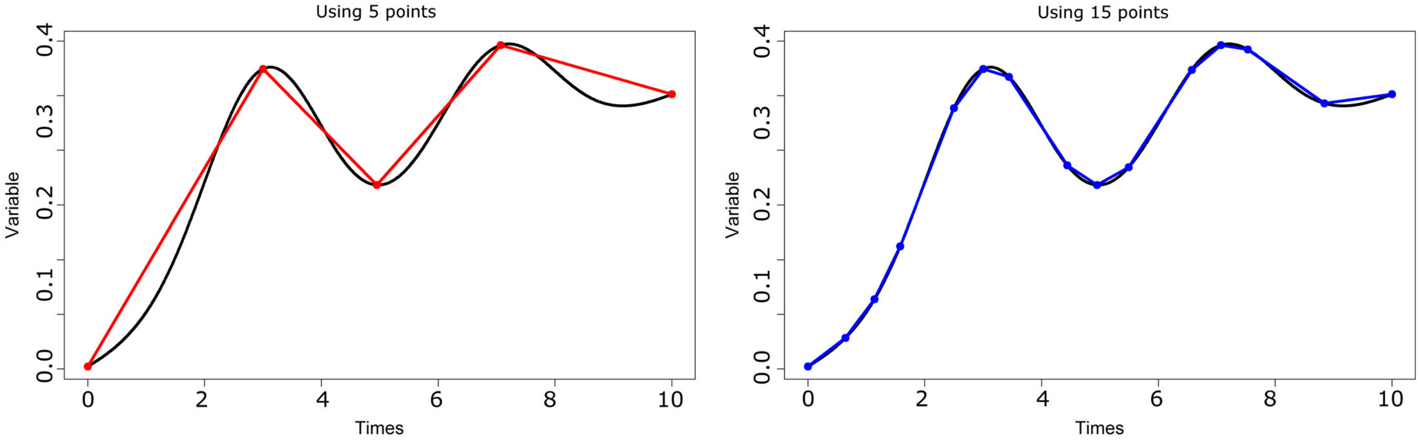


Fig 6. Approximation of a trajectory (in black). (a) using 5 points; (b) using 15 points.

doi:10.1371/journal.pone.0150738.g006

Peuker algorithm returns the greatest distance between the simplified curve and the initial curve. Thus, if the user has no direct control on the error, he/she has an estimation of it. He/She can then decide whether the size of the error seems too big for him/her to increase the number of points used by the Douglas-Pecker algorithm. The user may feel also free to use the classical Douglas-Pecker algorithm (control the error but not the number of points). In the latter case, the time complexity of the algorithm kmlShape is not guaranteed.

2.4 Overall complexity

In the end, the election cost is $O(n_S\, n_t)$. The cost of senator simplification is $O(t_{DP}^2 n_S t)$. One may then use kmlShape with the n_S simplified senators at cost $O(n_S t_{DP}^2)$. The overall complexity is $O(n_S nt + t_{DP}^2 n_S t + n_S t_{DP}^2)$, t_{DP} and n_S being constants set by the user. So, the final complexity is $O(nt)$.

Fig 7 summarizes the steps needed to partition data using algorithm kmlShape in a reasonable time.

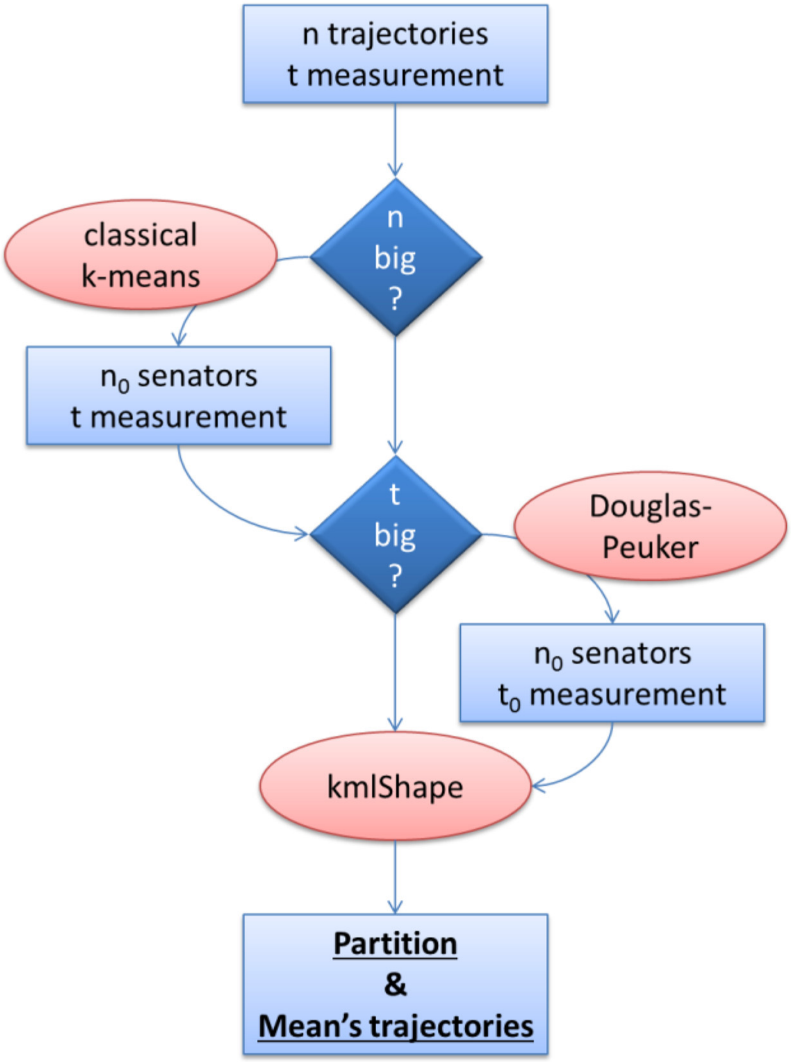


Fig 7. Steps for data partitioning with kmlShape the final complexity is in *O*(*nt*).

doi:10.1371/journal.pone.0150738.g007

3 Performance assessment

3.1 Simulation study

Generation of artificial data. Let us consider three populations of small ($n = 20$, $t = 21$), medium ($n = 40$, $t = 41$), and large size ($n = 500$, $t = 501$). Each population has k subgroups. Each subgroup G is defined by a typical trajectory $y = f_G(x)$. We have considered two cases:

- **Case 1**: two groups A and B (k = 2) with $f_A(x) = \psi(x, 0.5, 0.1) \times 0.125$ and $f_B(x) = \psi(x, 0.5, 0.1) \times 0.25$.
- **Case 2**: four groups A, B, C and D (k = 4) with $f_A(x) = \psi(x, 0.5, 0.1) \times 0.125$, $f_B(x) = \Phi(x, 0.4, 0.1) \times 0.5$, $f_C(x) = \psi(x, 0.5, 0.1) \times 0.25$ and $f_D(x) = \Phi(x, 0.4, 0.1)$.

with ψ the normal law distribution $\psi(x, m, s) = \frac{1}{s\sqrt{2\pi}} exp\left(-\frac{1}{2}\left(\frac{x-m}{s}\right)^2\right)$ and Φ its cumulative distribution function (see Fig 8).

Then each trajectory y_i belonging to subgroup G is a distortion of f_G. To create y_i, we choose a coefficient of distortion σ. Three types of distortion may be considered:

- **Simple distortion** it consists in a mere horizontal translation of f_G: $y_{ij} = f_G(x_{ij} + b_1)$, with $b_1 \sim \mathcal{U}(-\sigma, +\sigma)$
- **Multiple distortion** it consists not only in vertical and horizontal translations but also in vertical and horizontal deformations (compression and stretching): $y_{ij} = a_2.f_G(a_1.x_{ij} + b_1) + b_2$, with $(a_1, a_2) \sim \mathcal{U}(1-\sigma, 1+\sigma)^2$ and $(b_1, b_2) \sim \mathcal{U}(-\sigma, +\sigma)^2$.
- **Noisy distortion** it consists in a multiple distortion (as described above) together with a Gaussian random noise: $y_{ij} = a_2.f_G(a_1.x_{ij} + b_1) + b_2 + e_{ij}$, with $(a_1, a_2) \sim \mathcal{U}(1-\sigma, 1+\sigma)^2$, $(b_1, b_2) \sim \mathcal{U}(-\sigma, +\sigma)^2$ and $e_{ij} \sim \mathcal{N}(0, \sigma)$.

with $\mathcal{U}(a, b)$ the uniform distribution with minimum a and maximum b. For each type of distortion, σ takes values in {0.05, 0.1, 0.25}. At the end, (3 possible populations) times (2 possible cases) times (3 possible distortions) times (3 possible σ) gives 54 possible datasets. Each dataset was generated 500 times.

Small datasets were partitioned using (1.a) kmlShape (methods randomAll, $\lambda = 0.1$), (1.b) kmlShape using DTW (methods randomAll, $\lambda = 0$), (1.d) classical Euclidian k-means, and (1.e)

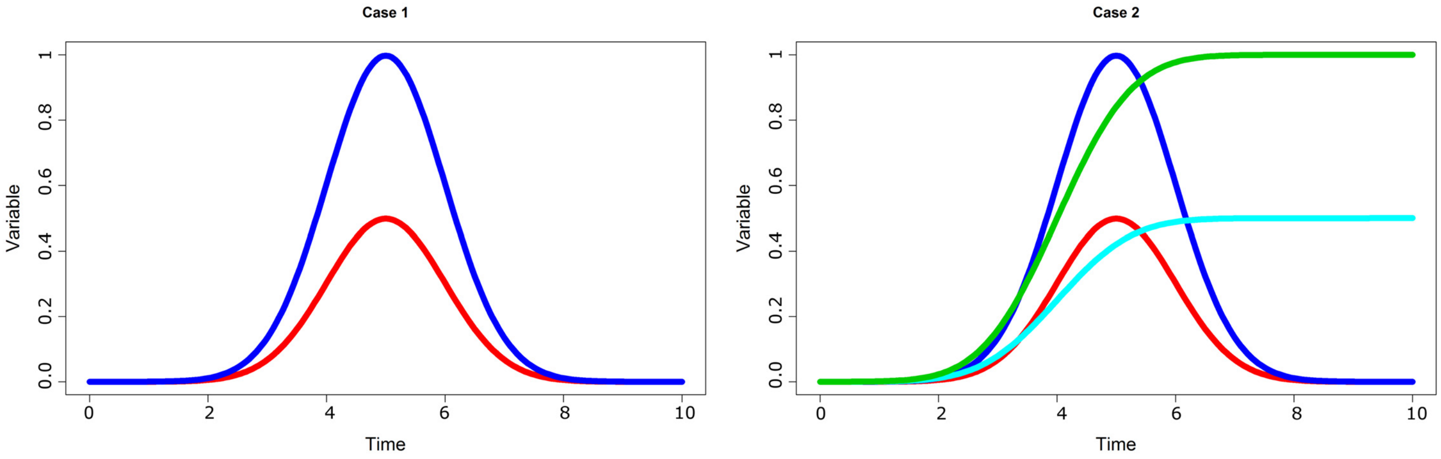


Fig 8. Artificial data. (a) Case 1: f_A is in red, f_B in green; (b) Case 2: f_A is in red, f_B in green, f_C in deep blue and f_D in light blue.

doi:10.1371/journal.pone.0150738.g008

fdakma [27, 31]. Medium datasets were partitioned using (2.a) kmlShape (randomAll, $\lambda = 0.1$), (2.b) kmlShape using DTW (randomAll, $\lambda = 0$), (2.c) kmlShape with simplification ($n_S = 32$, $t_{DP} = 21$, randomAll, $\lambda = 0.1$), (2.d) classical Euclidian k-means and (2.e) fdakma.

Large datasets were partitioned using (3.c) kmlShape with simplification ($n_S = 32$, $t_{DP} = 21$, randomAll, $\lambda = 0.1$) and (3.d) classical Euclidian k-means; the other methods were too time-consuming.

The study of (1.a, 1.b, 1.d, 1.e, 2.a, 2.b, 2.c, 2.d and 2.e) will allow us to compare kmlShape using Fréchet, kmlShape using DTW, classical k-means and fdakma in various conditions. The comparison of (2.a) and (2.c) will allow us to study the impact of the simplification procedures. (3.c) and (3.d) will allow the comparison of kmlShape with simplification and k-means performance on large data set.

The indicated parameters were chosen because they reflect an equilibrium between a slight deformation of the original data (that requires high n_S and t_{DP} values) and a reasonable calculation time (that requires low n_S and t_{DP} values).

3.2 Performance

To measure the performance, we used the Correct Classification Rate (cRate) which is the percentage of agreement between the partitioning found P and the true partitioning P_T. We have also used the adjusted Rand Index (aRand) [57] which is a variant of the Rand Index [58]; the cRand index being the proportion of pairs of individuals (i, j) who are either in the same cluster in P or in P_T or in separate clusters in P or in P_T. The adjusted rand index is simply aRand = (cRand – theoretical cRand) / (Maximum cRand – theoretical cRand). This adjustment makes the aRand take value 0 when it measures the agreement between two random partitions. These two measures of agreement between classifications have been already used by several authors [5, 9, 30, 59].

4 Results

4.1 Method comparisons

The respective performances of kmlShape, fdakma, and k-means with small and medium datasets, case 1 and 2, are shown Table 1. We observed that kmlShape performs better than fdakma and k-means regarding the classification indices. The same was found when only one specific subgroup was analyzed (e.g., only Case 1 with $\sigma = 0.05$). However, the differences between kmlShape and the other methods were more marked in Case 1 than in Case 2. Also, these differences tended to decrease slightly when σ increased.

Note that for these two examples, the results of kmlShape with $\lambda = 0.1$ and kmlShape using DTW (i.e. $\lambda = 0$) were identical. Thus we give only the results of kmlShape with $\lambda = 0.1$.

Table 1. Performance of classical k-means, fdakma, and kmlShape in case of small and medium datasets (mean value ± standard deviation).

	Classical k-means	kmlShape	fdakma
Case 1			
cRate	0.75 (± 0.12)	0.84 (± 0.15)	0.57 (± 0.07)
aRand	0.56 (± 0.2)	0.71 (± 0.26)	0.37 (± 0.1)
Case 2			
cRate	0.66 (± 0.14)	0.94 (± 0.12)	0.57 (± 0.07)
aRand	0.14 (± 0.27)	0.84 (± 0.32)	0.01 (± 0.06)

doi:10.1371/journal.pone.0150738.t001

Table 2. Performance of classical k-means, kmlShape, and simplified kmlShape with medium datasets (mean value ± standard deviation).

	Classical k-means	kmlShape	Simplified kmlShape
Case 1			
cRate	0.64 (± 0.13)	0.94 (± 0.12)	0.94 (± 0.12)
aRand	0.13 (± 0.25)	0.84 (± 0.31)	0.82 (± 0.32)
Case 2			
cRate	0.75 (± 0.11)	0.84 (± 0.15)	0.82 (± 0.14)
aRand	0.56 (± 0.19)	0.72 (± 0.25)	0.7 (± 0.24)

doi:10.1371/journal.pone.0150738.t002

4.2 Impact of simplification in terms of time or number of individuals

The respective performances of kmlShape, classical k-means, and simplified kmlShape with medium size data are shown Table 2. We observed that the performance of simplified kmlShape was close to (similar or slightly lower) the performance of kmlShape without simplification. In all cases, the performance of kmlShape was clearly better than that of a classical partitioning.

The performances of classical k-means and simplified kmlShape with large datasets are shown Table 3. The simplified kmlShape outperformed clearly the classical k-means. We also observed that the performance of kmlShape with simplification was quite close to that of kmlShape without simplification.

4.3 Application to real data

Cohort ICTUS. Ictus (see S1 File) is a cohort of 1380 patients with Alzheimer disease followed-up in 12 European countries [60, 61]. These patients were included between February 2003 and July 2005 in 29 centers specialized in neurology, geriatrics, psychiatry or psycho-geriatrics with a recognized experience in the diagnosis and management of Alzheimer disease. Most of these patients were seen during memory consultations and included consecutively. These patients were examined at six-month intervals over two years. Each examination included (though not exclusively) an Instrumental Activities of Daily Living (IADL) assessment.

A classical analysis of IADL trajectories using either mixture models or k-means revealed 4 groups (Fig 9.a). The main feature of these groups is to show close consistent declines. Using kmlShape (after using the data size reduction $n_s = 128$, no curve simplification) with 4 groups (Fig 9.b), three of these groups were similar to those found by other classical algorithms whereas a fourth "rapid decline" group was detected by kmlShape only.

The identification of this group makes it possible to predict and anticipate the needs of these patients in terms of informal help or professional care. Such a planning is of utmost

Table 3. Performances of classical k-means and simplified kmlShape with large data (mean value ± standard deviation).

	Classical k-means	Simplified kmlShape
Case 1		
cRate	0.59 (± 0.12)	0.92 (± 0.15)
aRand	0.09 (± 0.22)	0.79 (± 0.38)
Case 2		
cRate	0.71 (± 0.11)	0.8 (± 0.15)
aRand	0.56 (± 0.18)	0.68 (± 0.23)

doi:10.1371/journal.pone.0150738.t003

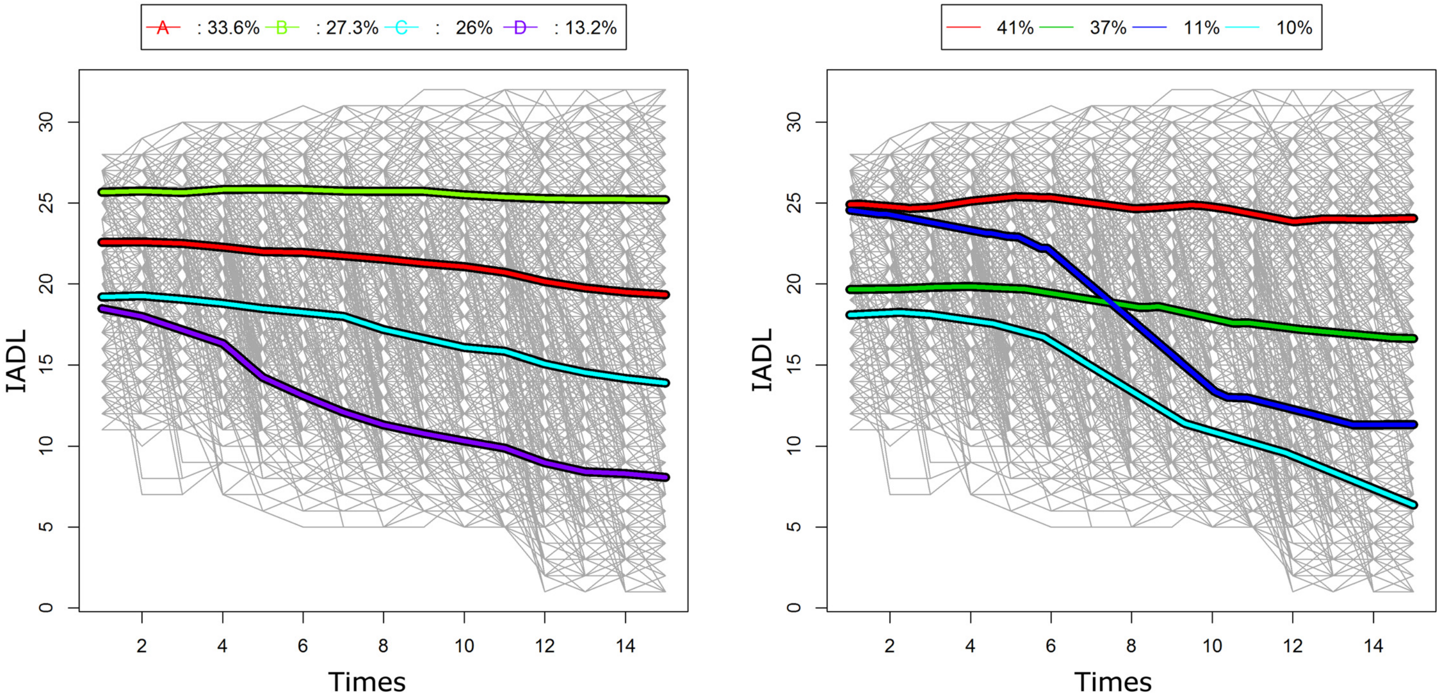


Fig 9. IADL trajectories, in 4 clusters. (a) with a classical method; (b) with kmlShape. kmlShape is able to identify a "rapid decline" cluster that is not be found using the classical method.

doi:10.1371/journal.pone.0150738.g009

importance for the health professionals and the families. Applying the same methods on other functions of these patients (cognition, behavior) would help clarifying the natural history of the disease. Applying these methods before diagnosis would help targeting the population to include in clinical trials for the prevention of Alzheimer disease. Indeed, such trials may use aggressive agents (e.g., monoclonal antibodies) which makes it necessary, from an ethical point of view, to target only the subpopulation with "rapid decline", which is not possible with the classical classification methods.

Quidel database. The QUIDEL database aims to gain better knowledge on hormone profiles of women without fertility problems. This database has been described as the largest existing database on hormone profiles in normally menstruating women and includes ovary ultrasound scans on the day of ovulation [62]. The database includes 107 women and 283 cycles with identification of the day of ovulation and daily titrations of the levels of the four main hormones of the ovulation cycle. It has been already the subject of publications [63, 64]

The use of classical classification methods regarding the luteinizing hormone (LH) provided three typical trajectories with similar features but with slight shifts in time (Fig 10.a). This is a typical finding in medicine but is currently strongly questioned [62, 65]. The use of kmlShape (after using the data size reduction $n_s = 128$, $t_{DP} = 20$) with three groups led to identifying: i) a classical profile that would concern 25% of the women; ii) a two LH-peak profile that would concern 22% of the women; and, iii) a profile with a single peak followed by a slow decline over several days that would concern the remaining 53% of the women (Fig 10.b).

These results will enrich the current debate on the role of the LH peak in the maintenance of corpus luteum. Indeed, as indicated by its name, the LH was first described as luteinizing but, as LH peaks occur close to the day of ovulation, LH was made responsible for triggering ovulation. However, recent works [66] have demonstrated that the course of LH during the

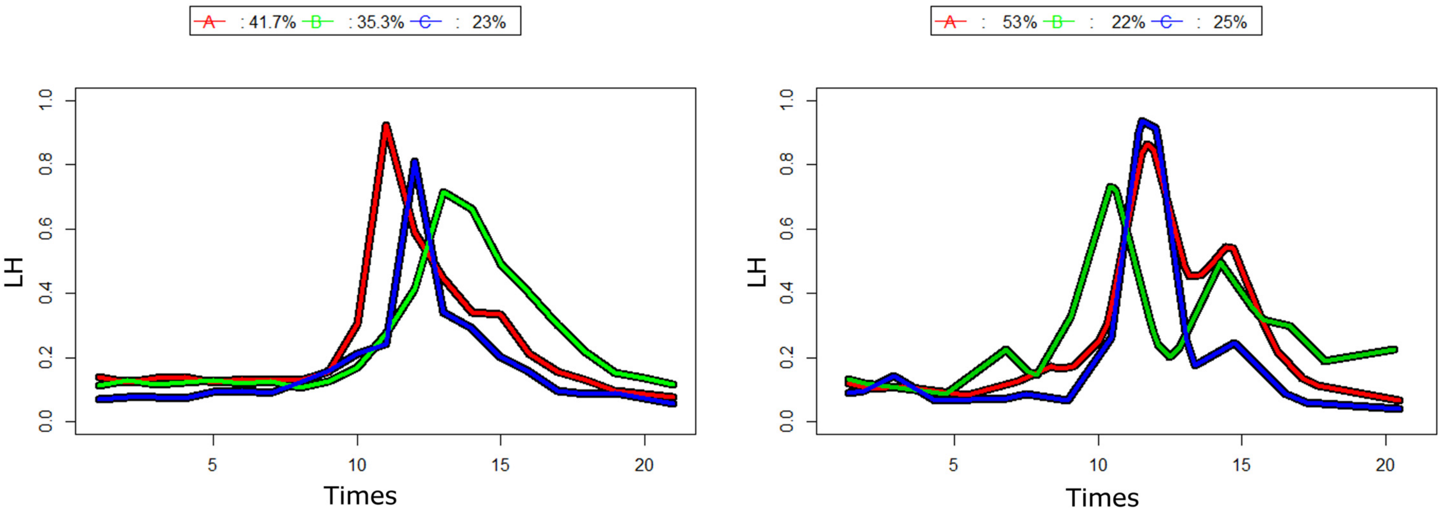


Fig 10. LH trajectories, in 3 clusters. (a) with a classical method; (b) with kmlShape. kmlShape shows typical trajectories with two peaks that are not found with the classical method.

doi:10.1371/journal.pone.0150738.g010

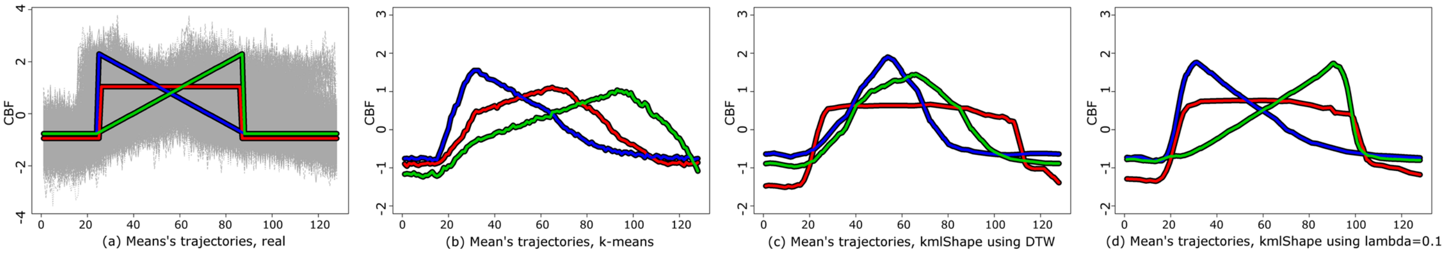
days that follow ovulation may be important to understand some abnormalities of corpus luteum and, thus, the implantation of the embryo in the uterus. The identification of 22% of women with LH double peak profiles is important for further research in reproductive biology.

UCR-CBF. The two last examples (CBF and Trace, see below) are extracted from the "UCR Time Series Classification Archive" [67], a collection of real and artificial datasets dedicated to studies of time series and longitudinal data. Contrarily to the approach we have adopted in our simulation study, these datasets are generated once.

CBF (Cylinder-Bell-Funnel, [68]) is a dataset of 900 trajectories measured 128 times. These trajectories are divided into three clusters of sizes 302, 300, and 298. The mean trajectories of each cluster are shown Fig 11.a. We partitioned the data with different methods:

- classical Euclidian k-means;
- kmlShape using DWT with simplification ($n_S = 64$, $t_{DP} = 30$, randomAll, $\lambda = 0$)
- kmlShape with simplification ($n_S = 64$, $t_{DP} = 30$, randomAll, $\lambda = 0.1$) and

The other methods were too time-consuming or did not converge. The classical classification methods found three groups with identical shapes shifted in time (Fig 11.b). kmlShape using DTW identified three groups with similar shapes but different heights (Fig 11.c). With



(a) Means's trajectories, real
(b) Mean's trajectories, k-means
(c) Mean's trajectories, kmlShape using DTW
(d) Mean's trajectories, kmlShape using lambda=0.1

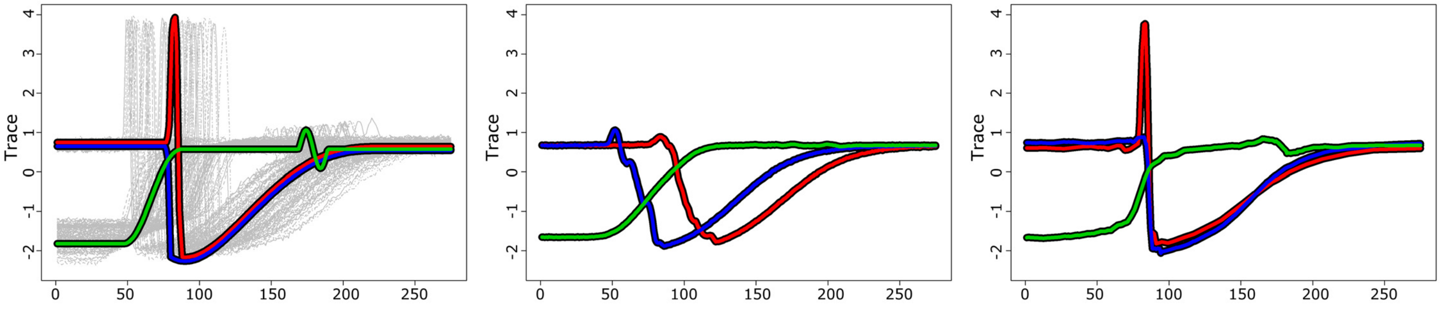
Fig 11. CBF trajectories. (a) real means (b) means found using a classical method; (c) means foung using kmlShape with DTW (d) means found using kmlShape with $\lambda = 0.1$. kmlShape found the real means.

doi:10.1371/journal.pone.0150738.g011

Table 4. Performance of classical k-means, simplified kmlShape using DTW and simplified kmlShape with $\lambda = 1$ on CBF dataset.

	Classical k-means	Simplified kmlShape with DTW	Simplified kmlShape with $\lambda = 1$
cRate	0.65	0.59	0.95
aRand	0.34	0.25	0.86

doi:10.1371/journal.pone.0150738.t004



(a) Mean's trajectories, real

(b) Mean's trajectories, k-means

(c) Mean's trajectories, kmlShape

Fig 12. Trace trajectories. (a) real means (b) means found using a classical method; (c) means found using kmlShape with $\lambda = 0.1$. kmlShape found the real means.

doi:10.1371/journal.pone.0150738.g012

these two classification methods, the number of misclassified trajectories was quite important (Table 4) and the average trajectories obtained were quite different from the average trajectories used to generate groups (Fig 11.a). kmlShape with $\lambda = 0.1$ gave good results in terms of individual ranking as in terms of identification of the average trajectory (Fig 11.d).

Note that, in this example, the use of the DTW method gave a different (worse) result than the Fréchet mean with $\lambda = 0.1$. The reasons for this will be discussed in the appendix A.

UCR-Trace. The Trace database [69] (also extracted from the "UCR Time Series Classification Archive") was obtained from EDF (Electricité de France). The database includes 200 different transient classes (Fig 12.a). We clustered the data using:

- classical Euclidian k-means;
- kmlShape using DWT with simplification ($n_S = 64$, $t_{DP} = 30$, randomAll, $\lambda = 0$)
- kmlShape with simplification ($n_S = 64$, $t_{DP} = 30$, randomAll, $\lambda = 0.1$) and

The other methods were too time-consuming or did not converge. kmlShape $\lambda = 0.1$ and kmlShape using DTW gave exactly the same results (we report here only kmlShape with $\lambda = 0.1$). The classical classification method identified perfectly one of the three groups (the group in blue Fig 12.b) but failed to distinguish between the two others. The kmlShape identified three groups without errors and found the right average trajectories (Fig 12.c and Table 5.)

Table 5. Performance of classical k-means, simplified kmlShape using DTW and simplified kmlShape with $\lambda = 0.1$ on Trace dataset.

	Classical k-means	Simplified kmlShape
cRate	0.77	1
eRand	0.73	1

doi:10.1371/journal.pone.0150738.t005

5 Discussion

In the present article, we introduce kmlShape, a novel method of data partitioning. This method provides clusters on the basis of trajectory shape. This allows especially grouping individuals whose trajectories have similar forms but shifted positions in time. Given the high algorithmic complexity of kmlShape, we present two data-simplification methods that allow reducing the lengths of the trajectories or the number of the individuals of the population under study.

In comparison with other shape-based partitioning methods, kmlShape demonstrated higher performances with the datasets tested whatever the variance, the population size, or the number of clusters. In addition, the search for loss of information due to data simplification has shown that the final partitioning is only slightly affected by this simplification (Table 2). Another advantage was that each of the partition steps (reduction of the number of measurements or of the population size) is mathematically simple, graphically displayable, and easily checkable. At any moment, the user can decide to invalidate any excessive "data simplification". Finally, with real datasets, kmlShape makes it possible to detect groups of individuals of non-negligible sizes that would not be detected by other classical methods. Thus, the method paves the way to new perspectives in terms of data analysis.

Within the general context of data partitioning, the problem of the optimal number of clusters is still an open issue. Numerous criteria exist, either parametric (BIC [70], AIC [71], AICc [72], global posterior probability [73], . . .) or non-parametric (Caliskin & Harabatz [74], Ray & Turi [75], Davies & Bouldin [76], . . .). These criteria are regularly compared using artificial data [77, 78]. With real data, they often suffer from bias. For example, Calinski & Harbatz criteria (which is the best criterion according to both [77] and [78]) often select the lowest number of clusters. Also, different authors advise to choose the number of clusters on the basis of clinical relevance rather than an index [13].

In the case of partitioning using the Fréchet distance, the problem is more complicated because the classical criteria are designed to be used with classical distances. To date, there is no quality criterion that can help selecting the correct number of clusters within the context of respecting-shape partitioning. Finding such a quality index would be a non-negligible progress in the field of data partitioning.

Regarding the choices of n_S and t_{DP}, the present study showed that $n_S = 32$ and $t_{DP} = 20$ is a good compromise between a reasonable simplification and an acceptable calculation time. Obviously, these parameters may be adapted according to the type of data (with complex and long curves, $t_{DP} = 20$ seems to be insufficient; with simple curves $t_{DP} = 10$ may be sufficient) and the power of the computer involved. In a medium term, new high-performance statistical software programs will probably overcome the current limitations.

The choice of λ is more complex. As shown Fig 3, it changes the relative weight of the distance between two trajectories according to the x-axis and the y-axis. If the x-scale and the y-scale are identical, setting $\lambda = 0.1$ gives ten times more weight to a vertical offset than to a horizontal offset. This case is close to the one shown in the right panel Fig 3: i_1 is close to i_3 because the "horizontal offset" is very important. When λ is 1, the horizontal and the vertical offsets have the same importance. When $\lambda = +\infty$, the horizontal offsets becomes very expensive, the Fréchet distance is then identical to the classical maximum distance. When $\lambda = 0$, the horizontal offsets becomes free, the Fréchet distance is then identical to the dynamic time warping distance. With our artificial examples, a value of $\lambda = 0.1$ or less allowed a correct identification of the groups. More detail about λ can be find appendix A.

When the scales are not the same (which is true in the majority of cases in the present study because one axis represents time and the other the variable of interest), the data can be standardized by dividing by the range of x and multiplying by the range of y. On our real examples, the value used was $\lambda = \frac{Max(y_{it})-Min(y_{ij})}{Max(j)-Min(j)} \times 0.1$. This value is the one that gave the most relevant results from the clinical point of view and has identified groups undetected by conventional techniques.

A How to choose λ?

The choice of λ is of considerable impact on the final clustering. There is no "best" value for λ as there is no "best clustering techniques ever", it must be selected depending on the problem. For that, according to the specific problem, the user has to define the curves that should be considered as "close curves". To make this decision, it is important to keep in mind the principle of working of λ.

Consider Fig 13.a. The black trajectory represents an individual i. The three colored trajectories represent three cluster centers. The question is to decide which i should be the closest. Suppose that the trajectories represent the intensity of a disease. From a public health perspective, it is important to know when the vaccines should be ready so it is interesting to group i with A. For a researcher who wants to understand the disease, the type of disease progression is more important than the time of its outbreak so it will be more relevant to group i and C because. In some other problems, it might be interesting to group i and B.

Consider now the Fréchet distance between i and the cluster centers (represented by the segments between the curves in dash). In this example, i is close to the average trajectory B (the dashed blue line is shortest than the green or the red lines). If we represent the same data but divide the scale of the x axis by two (Fig 13.b), i is closer to C (the green dashed line is the shortest). If we represent the same data but multiply the scale of the x by 2 (Fig 13.c), i is close to A (the red dashed line is the shortest).

From a mathematical perspective, this is easy to understand: the calculation of the length of a segment involves two components; the differences in values along axis x and along axis y. Changing the scale of x changes the relative importance of the two differences (reducing the x' scale decreases the importance of the difference along the axis of x). In Fig 13.a, a horizontal shift has a great impact on the calculation of the distance. So i is close to A. In Fig 13.b, a horizontal shift has a little impact on the distance calculation. So i is close to C. In extreme cases, when λ tends to $+\infty$, a small difference on the x-axis increases greatly the distance between the curves. The Fréchet distance is reached when "man and dog" keep the same abscissa at any point (as a difference of δ causes an increase in the distance of $\delta \times \lambda$ which tends to $+\infty$ as λ tends to $+\infty$). The Fréchet distance is then identical to the max distance.

Conversely, when λ is 0, the distance between points (x_1, y_1) and (x_2, y_2) is just $|y_1 - y_2|$. The difference along the x-axis has no impact on the distance between the curves. The Fréchet distance matches the DTW.

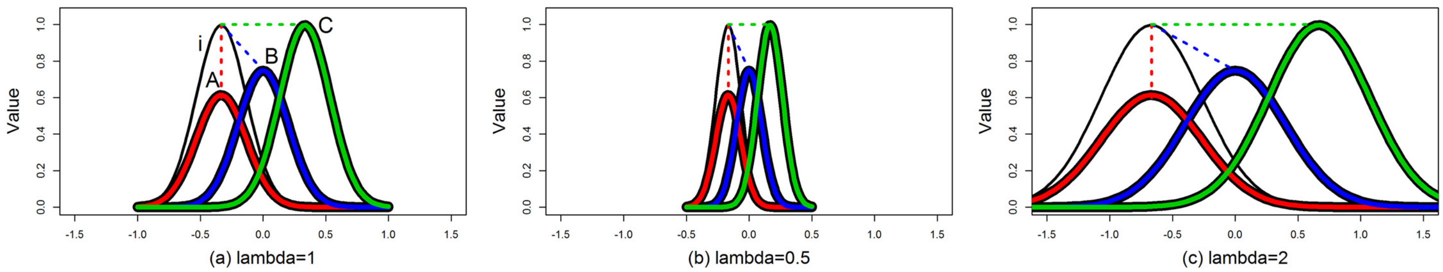


Fig 13. The impact of λ on the distance between the trajectory *i* and the clusters mean's trajectories *A*, *B* and *C*.

doi:10.1371/journal.pone.0150738.g013

In summary, the role of λ is to allow the user to choose the case in which he wishes to be. Suppose that the initial population is shown Fig 13.b. If, according to the problem, it is relevant to cluster i with A, then λ should be small ($\lambda < <1$). If i should be clustered with B, then $\lambda = 1$ will be a correct choice. If i should be close to C, then λ should be big ($\lambda > >1$).

B Appendix: Computational complexities

B.1 Euclidean distance

The formula for calculating the Euclidean distance is $Dist(y_{i.}, y_{i'.}) = \sum_{j=1}^{t} (y_{ij} - y_{i'j})^2$, that is subtraction and a square for each j, then $t - 1$ additions. The overall complexity is $O(t)$.

B.2 The Fréchet distance

The calculation of the Fréchet distance needs the calculation of the distance matrix between each pair of points $\left(\begin{pmatrix} j \\ y_{ij} \end{pmatrix}, \begin{pmatrix} j' \\ y_{i',j'} \end{pmatrix}\right)$. This is a matrix of size t^2. The complexity of the calculation of the distance between each pair of points is a constant, so the complexity of the Fréchet distance is $O(t^2)$.

The computing complexity of the Fréchet path is identical because, in addition to the computation of the matrix distance between each pair of points, it only requires browsing the matrix once to find the path.

B.3 The Fréchet mean between two trajectories

The computing complexity of the Fréchet mean between two trajectories needs the calculation of the Fréchet path (cost: $O(t^2)$). The length of the Fréchet path is bounded by $2t - 2$. Then the computation of the mean needs up to $2t - 2$ additions and divisions. Thus the overall complexity is $O(t^2)$.

B.4 The Fréchet mean between two trajectories

The generalization of the Fréchet mean to n trajectories requires the calculation of an index for each tuple $\left(\begin{pmatrix} j_1 \\ y_{i_1 j_1} \end{pmatrix}, \begin{pmatrix} j_2 \\ y_{i_2,j_2} \end{pmatrix}, \ldots, \begin{pmatrix} j_n \\ y_{i_n,j_n} \end{pmatrix}\right)$, that is, a matrix of size t^n. The complexity of the calculation is therefore at least $O(t^n)$.

B.5 The Fréchet mean, method RandomAll

Method RandomAll merges n individuals two by two. It takes $n - 1$ merges (cost: $O(t^2)$). The overall complexity is $O(nt^2)$.

B.6 The Fréchet mean, method Hierarchical

Method Hierarchical computes the Fréchet distance (cost: $O(t^2)$) between all possible couples ($n(n - 1)/2$ couples) for a total cost of $O(n^2\, t^2)$. Then, it merges the n individuals two by two. Each merging has a cost of $O(t^2)$. The final complexity is $O(n^2\, t^2)$.

B.7 The Fréchet mean, method RandomSubset

Method RandomAll merges n_0 individuals two by two. It takes $n_0 - 1$ merges. Each merge costs $O(t^2)$. The overall complexity is $O(n_0\, t^2)$.

B.8 k-means

At each stage of k-means, nk Euclidean distances (cost: $O(t)$) between n individuals and k groups centers are calculated (cost: $O(tnk)$). Then, in each group g, the mean of the n_g individuals belonging to the group is calculated; that is, tn_g additions per group with $n = \sum n_g$. The final complexity is $O(tnk)$ (the number of iterations is neglected here because it is generally bounded, it is therefore a constant).

B.9 kmlShape

At each stage of k-means, nk Fréchet distances (complexity $O(t^2)$) between n individuals and k groups centers are calculated (cost: $O(t^2\ nk)$). Then, in each group g, the mean of n_g individuals in the group is calculated. The cost is $O(n_g\ t^2)$ per group with method RandomAll, $O(n_g^2 t^2)$ with method hierarchical. The final complexity is $O(knt^2)$ with method RandomAll and $O(n^2\ t^2)$ with method Hierarchical.

B.10 Douglas-Peuker algorithm

For a curve which must be simplified into t_0 points, each iteration requires the calculation of the distance between the t points and the current curve (cost: $O(t.t_0)$). This has to be done t_0 times. So, for n curves, the complexity is $O(ntt_0^2)$.

Supporting Information

S1 File. A subset of the longitudinal study ICTUS.
(CSV)

Acknowledgments

The authors thank Jean Iwaz (Hospices Civils de Lyon) for the revision of the manuscript draft. We also thank the reveiwers whose careful readings and insighful comments allowed us to greatly improve the quality of this article.

Funding: This research was founded by the ANR grant IDoL: ANR-12-BSV1-0036.

Author Contributions

Conceived and designed the experiments: CG RE TD. Performed the experiments: CG MB. Analyzed the data: CG. Contributed reagents/materials/analysis tools: RE SA. Wrote the paper: CG RE TD SA FS MB.

References

1. Tarpey T, Kinateder KKJ. Clustering Functional Data. Journal of Classification. 2003; 20(1):93–114. doi: 10.1007/s00357-003-0007-3
2. Garcia-Escudero LA, Gordaliza A. A proposal for robust curve clustering. Journal of classification. 2005; 22(2):185–201. doi: 10.1007/s00357-005-0013-8
3. Tarpey T. A parametric k-means algorithm. Computational statistics. 2007; 22(1):71–89. doi: 10.1007/s00180-007-0022-7 PMID: 17917692
4. Elsensohn MH, Klich A, Ecochard R, Bastard M, Genolini C, Etard JF, et al. A graphical method to assess distribution assumption in group-based trajectory models. Statistical methods in medical research. 2013;p. 0962280213475643.
5. Genolini C, Pingault J, Driss T, Côté S, Tremblay RE, Vitaro F, et al. KmL3D: a non-parametric algorithm for clustering joint trajectories. Computer methods and programs in biomedicine. 2013; 109 (1):104–111. doi: 10.1016/j.cmpb.2012.08.016 PMID: 23127283

6. Lee JG, Han J, Whang KY. Trajectory clustering: a partition-and-group framework. In: Proceedings of the 2007 ACM SIGMOD international conference on Management of data. ACM; 2007. p. 593–604.

7. James GM, Sugar CA. Clustering for sparsely sampled functional data. Journal of the American Statistical Association. 2003; 98(462):397–408. doi: 10.1198/016214503000189

8. Luan Y, Li H. Clustering of time-course gene expression data using a mixed-effects model with B-splines. Bioinformatics. 2003; 19(4):474–482. doi: 10.1093/bioinformatics/btg014 PMID: 12611802

9. Chiou JM, Li PL. Functional clustering and identifying substructures of longitudinal data. Journal of the Royal Statistical Society: Series B (Statistical Methodology). 2007; 69(4):679–699. doi: 10.1111/j.1467-9868.2007.00605.x

10. Nagin DS. Analyzing developmental trajectories: a semiparametric, group-based approach. Psychological methods. 1999; 4(2):139. doi: 10.1037/1082-989X.4.2.139

11. Muthén B, Shedden K. Finite mixture modeling with mixture outcomes using the EM algorithm. Biometrics. 1999; 55(2):463–469. doi: 10.1111/j.0006-341X.1999.00463.x PMID: 11318201

12. Magidson J, Vermunt JK. Latent class models for clustering: A comparison with K-means. Canadian Journal of Marketing Research. 2002; 20:37.

13. Everitt B, Landau S, Leese M. Cluster Analysis. 4th. Arnold, London; 2001.

14. Fréchet MM. Sur quelques points du calcul fonctionnel. Rendiconti del Circolo Matematico di Palermo (1884–1940). 1906; 22(1):1–72. doi: 10.1007/BF03018603

15. Lucero JC, Munhall KG, Gracco VL, Ramsay JO. On the registration of time and the patterning of speech movements. Journal of Speech, Language, and Hearing Research. 1997; 40(5):1111–1117. doi: 10.1044/jslhr.4005.1111 PMID: 9328881

16. Al-Naymat G, Chawla S, Taheri J. SparseDTW: a novel approach to speed up dynamic time warping. In: Proceedings of the Eighth Australasian Data Mining Conference-Volume 101. Australian Computer Society, Inc.; 2009. p. 117–127.

17. Berndt DJ, Clifford J Seattle WA. Using Dynamic Time Warping to Find Patterns in Time Series. KDD workshop. 1994; 10(16):359–370.

18. Keogh E, Ratanamahatana CA. Exact indexing of dynamic time warping. Knowledge and information systems. 2005; 7(3):358–386. doi: 10.1007/s10115-004-0154-9

19. Vlachos M, Kollios G, Gunopulos D. Discovering similar multidimensional trajectories. In: Data Engineering, 2002. Proceedings. 18th International Conference on. IEEE; 2002. p. 673–684.

20. Chen L, Özsu MT, Oria V. Robust and fast similarity search for moving object trajectories. In: Proceedings of the 2005 ACM SIGMOD international conference on Management of data. ACM; 2005. p. 491–502.

21. Buchin K, Buchin M, Van Kreveld M, Luo J. Finding long and similar parts of trajectories. Computational Geometry. 2011; 44(9):465–476. doi: 10.1016/j.comgeo.2011.05.004

22. Ramsay J, Li X. Curve registration. Journal of the Royal Statistical Society: Series B (Statistical Methodology). 1998; 60(2):351–363. doi: 10.1111/1467-9868.00129

23. Ramsay J, Silvermann B. Functional Data Analysis. Springer Series in Statistics. Wiley Online Library; 1998.

24. Dimeglio C, Gallón S, Loubes JM, Maza E. A robust algorithm for template curve estimation based on manifold embedding. Computational Statistics & Data Analysis. 2014; 70:373–386. doi: 10.1016/j.csda.2013.09.030

25. James GM, et al. Curve alignment by moments. The Annals of Applied Statistics. 2007; 1(2):480–501. doi: 10.1214/07-AOAS127

26. Kaziska D, Srivastava A. Gait-based human recognition by classification of cyclostationary processes on nonlinear shape manifolds. Journal of the American Statistical Association. 2007; 102(480):1114–1124. doi: 10.1198/016214507000000464

27. Sangalli LM, Secchi P, Vantini S, Vitelli V. K-mean alignment for curve clustering. Computational Statistics & Data Analysis. 2010; 54(5):1219–1233. doi: 10.1016/j.csda.2009.12.008

28. Chudova D, Gaffney S, Mjolsness E, Smyth P. Translation-invariant mixture models for curve clustering. In: Proceedings of the ninth ACM SIGKDD international conference on Knowledge discovery and data mining. ACM; 2003. p. 79–88.

29. Gaffney S, Smyth P. Joint probabilistic curve clustering and alignment. In: Advances in neural information processing systems; 2004. p. 473–480.

30. Liu X, Yang MC. Simultaneous curve registration and clustering for functional data. Computational Statistics & Data Analysis. 2009; 53(4):1361–1376. doi: 10.1016/j.csda.2008.11.019

31. Patriarca M, Sangalli L, Secchi P, Vantini S, Vitelli V. fdakma: Clustering and alignment of a functional dataset; 2013. R package version 1.0. Available from: http://CRAN.R-project.org/package=fdakma.

32. Celeux G, Govaert G. A Classification EM algorithm for Clustering and Two Stochastic Versions. Computational Statistics & Data Analysis. 1992; 14(3):315–332. doi: 10.1016/0167-9473(92)90042-E

33. Hartigan JA, Wong MA. Algorithm AS 136: A K-means Clustering Algorithm. Journal of the Royal Statistical Society Series C (Applied Statistics). 1979; 28(1):100–108.

34. MacQueen J, et al.; USA California. Some methods for classification and analysis of multivariate observations. Proceedings of the fifth Berkeley symposium on mathematical statistics and probability. 1967; 1(281–297):14.

35. Genolini C, Falissard B. kml: k-means for Longitudinal Data. Computational Statistics. 2010; 25 (2):317–328. doi: 10.1007/s00180-009-0178-4

36. Divoux A, Tordjman J, Lacasa D, Veyrie N, Hugol D, Aissat A, et al. Fibrosis in human adipose tissue: composition, distribution, and link with lipid metabolism and fat mass loss. Diabetes. 2010; 59 (11):2817–2825. doi: 10.2337/db10-0585 PMID: 20713683

37. Pingault JB, Côté SM, Lacourse E, Galéra C, Vitaro F, Tremblay RE. Childhood hyperactivity, physical aggression and criminality: a 19-year prospective population-based study. PloS one. 2013; 8(5): e62594. doi: 10.1371/journal.pone.0062594 PMID: 23658752

38. Pingault JB, Tremblay RE, Vitaro F, Carbonneau R, Genolini C, Falissard B, et al. Childhood trajectories of inattention and hyperactivity and prediction of educational attainment in early adulthood: a 16-year longitudinal population-based study. American Journal of Psychiatry. 2011; 168(11):1164–1170. doi: 10.1176/appi.ajp.2011.10121732 PMID: 21799065

39. Mackelprang RD, Baeten JM, Donnell D, Celum C, Farquhar C, de Bruyn G, et al. Quantifying Ongoing HIV-1 Exposure in HIV-1–Serodiscordant Couples to Identify Individuals With Potential Host Resistance to HIV-1. Journal of Infectious Diseases. 2012; 206(8):1299–1308. doi: 10.1093/infdis/jis480 PMID: 22926009

40. Rancière F, Nikasinovic L, Bousquet J, Momas I. Onset and persistence of respiratory/allergic symptoms in preschoolers: new insights from the PARIS birth cohort. Allergy. 2013; 68(9):1158–1167. PMID: 23919292

41. Pena JM, Lozano JA, Larranaga P. An Empirical Comparison of Four Initialization Methods for the k-Means Algorithm. Pattern recognition letters. 1999; 20(10):1027–1040. doi: 10.1016/S0167-8655(99)00069-0

42. Khan SS, Ahmad A. Cluster Center Initialization Algorithm for k-means Clustering. Pattern Recognition Letters. 2004; 25(11):1293–1302. doi: 10.1016/j.patrec.2004.04.007

43. Redmond SJ, Heneghan C. A Method for Initialising the k-means Clustering Algorithm Using *kd*-trees. Pattern Recognition Letters. 2007; 28(8):965–973. doi: 10.1016/j.patrec.2007.01.001

44. Steinley D, Brusco MJ. Initializing k-means Batch Clustering: A Critical Evaluation of Several Techniques. Journal of Classification. 2007; 24(1):99–121. doi: 10.1007/s00357-007-0003-0

45. Twisk J, De Vente W. Attrition in Longitudinal Studies: How to Deal With Missing Data. Journal of Clinical Epidemiology. 2002; 55(4):329–337. doi: 10.1016/S0895-4356(01)00476-0 PMID: 11927199

46. Engels JM, Diehr P. Imputation of Missing Longitudinal Data: A Comparison of Methods. Journal of Clinical Epidemiology. 2003; 56(10):968–976. doi: 10.1016/S0895-4356(03)00170-7 PMID: 14568628

47. Genolini C, Écochard R, Jacqmin-Gadda H. Copy Mean: A New Method to Impute Intermittent Missing Values in Longitudinal Studies. Open Journal of Statistics. 2013; 3:26. doi: 10.4236/ojs.2013.34A004

48. Genolini C, Lacombe A, Écochard R, Subtil F. CopyMean: a new method to predict monotone missing values in longitudinal studies. Computer Methodes and Programs in biomedicine. 2015;In press:1–22.

49. Alt H, Godau M. Computing the Fréchet distance between two polygonal curves. International Journal of Computational Geometry & Applications. 1995; 5(01n02):75–91. doi: 10.1142/S0218195995000064

50. Keogh E, Chu S, Hart D, Pazzani M. An online algorithm for segmenting time series. In: Data Mining, 2001. ICDM 2001, Proceedings IEEE International Conference on. IEEE; 2001. p. 289–296.

51. Panagiotakis C, Pelekis N, Kopanakis I, Ramasso E, Theodoridis Y. Segmentation and sampling of moving object trajectories based on representativeness. Knowledge and Data Engineering, IEEE Transactions on. 2012; 24(7):1328–1343. doi: 10.1109/TKDE.2011.39

52. Cao H, Wolfson O, Trajcevski G. Spatio-temporal data reduction with deterministic error bounds. The VLDB Journal The International Journal on Very Large Data Bases. 2006; 15(3):211–228. Line simplification. doi: 10.1007/s00778-005-0163-7

53. Gudmundsson J, Katajainen J, Merrick D, Ong C, Wolle T. Compressing spatio-temporal trajectories. Computational geometry. 2009; 42(9):825–841. Amelioration de Douglas-peker. doi: 10.1016/j.comgeo.2009.02.002

54. Douglas DH, Peucker TK. Algorithms for the reduction of the number of points required to represent a digitized line or its caricature. Cartographica: The International Journal for Geographic Information and Geovisualization. 1973; 10(2):112–122. doi: 10.3138/FM57-6770-U75U-7727

55. Ramer U. An iterative procedure for the polygonal approximation of plane curves. Computer Graphics and Image Processing. 1972; 1(3):244–256. Douglas-pecker, sous un autre nom. doi: 10.1016/S0146-664X(72)80017-0

56. Duda RO, Hart PE, et al. Pattern classification and scene analysis. vol. 3. Wiley New York; 1973. Douglas-pecker, sous un autre nom.

57. Hubert L, Arabie P. Comparing partitions. Journal of classification. 1985; 2(1):193–218. doi: 10.1007/BF01908075

58. Rand WM. Objective criteria for the evaluation of clustering methods. Journal of the American Statistical association. 1971; 66(336):846–850. doi: 10.1080/01621459.1971.10482356

59. Genolini C, Alacoque X, Sentenac M, Arnaud C. kml and kml3d: R Packages to Cluster Longitudinal Data. Journal of Statistical Software. 2015; 65(4):1–34. Available from: http://www.jstatsoft.org/v65/i04/ . doi: 10.18637/jss.v065.i04

60. Reynish E, Cortes F, Andrieu S, Cantet C, Olde Rikkert M, Melis R, et al. The ICTUS Study: A prospective longitudinal observational study of 1,380 AD patients in Europe. Neuroepidemiology. 2007; 29(1–2):29–38.

61. Vellas B, Hausner L, Frolich L, Cantet C, Gardette V, Reynish E, et al. Progression of Alzheimer disease in Europe: Data from the European ICTUS study. Current Alzheimer Research. 2012; 9(8):902–912. doi: 10.2174/156720512803251066 PMID: 22742853

62. Ecochard R. Heterogeneity in Fecundability Studies: Issues and Modelling. Statistical Methods in Medical Research. 2006; 15(2):141–160. doi: 10.1191/0962280206sm436oa PMID: 16615654

63. Ecochard R, Gougeon A. Side of Ovulation and Cycle Characteristics in Normally Fertile Women. Human Reproduction. 2000; 15(4):752–755. doi: 10.1093/humrep/15.4.752 PMID: 10739814

64. Ecochard R, Boehringer H, Rabilloud M, Marret H. Chronological Aspects of Ultrasonic, Hormonal, and Other Indirect Indices of Ovulation. BJOG: An International Journal of Obstetrics & Gynaecology. 2001; 108(8):822–829. doi: 10.1111/j.1471-0528.2001.00194.x

65. Alliende ME. Mean versus individual hormonal profiles in the menstrual cycle. Fertility and sterility. 2002; 78(1):90–95. doi: 10.1016/S0015-0282(02)03167-9 PMID: 12095496

66. Direito A, Bailly S, Mariani A, Ecochard R. Relationships between the luteinizing hormone surge and other characteristics of the menstrual cycle in normally ovulating women. Fertility and sterility. 2013; 99 (1):279–285. doi: 10.1016/j.fertnstert.2012.08.047 PMID: 22999798

67. Chen Y, Keogh E, Hu B, Begum N, Bagnall A, Mueen A, et al. The UCR Time Series Classification Archive; 2015. www.cs.ucr.edu/∼eamonn/time_series_data/.

68. Kadous MW. Learning Comprehensible Descriptions of Multivariate Time Series. In: ICML; 1999. p. 454–463.

69. Roverso D. Multivariate temporal classification by windowed wavelet decomposition and recurrent neural networks. In: 3rd ANS international topical meeting on nuclear plant instrumentation, control and human-machine interface. vol. 20; 2000. p. 1–8.

70. Schwarz G. Estimating the Dimension of a Model. The Annals of Statistics. 1978; 6(2):461–464. doi: 10.1214/aos/1176344136

71. Akaike H. A New Look at the Statistical Model Identification. Automatic Control, IEEE Transactions On. 1974; 19(6):716–723. doi: 10.1109/TAC.1974.1100705

72. Hurvich CM, Tsai CL. Regression and Time Series Model Selection in Small Samples. Biometrika. 1989; 76(2):297–307. doi: 10.1093/biomet/76.2.297

73. Bolstad WM. Introduction to Bayesian Statistics. John Wiley & Sons-Interscience; 2007. doi: 10.1002/9780470181188

74. Calinski T, Harabasz J. A dendrite method for cluster analysis. Communications in Statistics. 1974; 3 (1):1–27. doi: 10.1080/03610927408827101

75. Ray S, Turi RH. Determination of number of clusters in k-means clustering and application in colour image segmentation. In: Proceedings of the 4th International Conference on Advances in Pattern Recognition and Digital Techniques (ICAPRDT'99), Calcutta, India; 1999. p. 137–143.

76. Davies DL, Bouldin DW. A cluster separation measure. IEEE Transactions on Pattern Analysis and Machine Intelligence. 1979; 1(2):224–227. doi: 10.1109/TPAMI.1979.4766909 PMID: 21868852

77. Milligan GW, Cooper MC. An examination of procedures for determining the number of clusters in a data set. Psychometrika. 1985; 50(2):159–179. doi: 10.1007/BF02294245

78. Shim Y, Chung J, Choi IC. A Comparison Study of Cluster Validity Indices Using a Nonhierarchical Clustering Algorithm. In: Proceedings of CIMCA-IAWTIC'05-Volume 01. IEEE Computer Society Washington, DC, USA; 2005. p. 199–204.