Summarizing a Set of Time Series by Averaging: from Steiner Sequence to Compact Multiple Alignment

François Petitjean[a,*], Pierre Gançarski[a]

[a] LSIIT – UMR 7005, University of Strasbourg, 67412 Illkirch Cedex, France

Abstract

Summarizing a set of sequences is an old topic that has been revived in the last decade, due to the increasing availability of sequential datasets. The definition of a consensus object is on the center of data analysis issues, since it crystallizes the underlying organization of the data.

Dynamic Time Warping (DTW) is currently the most relevant similarity measure between sequences for a large panel of applications, since it makes it possible to capture temporal distortions. In this context, averaging a set of sequences is not a trivial task, since the average sequence has to be consistent with this similarity measure.

The Steiner theory and several works in computational biology have pointed out the connection between multiple alignments and average sequences. Taking inspiration from these works, we introduce the notion of compact multiple alignment, which allows us to link these theories to the problem of summarizing under time warping. Having defined the link between the multiple alignment and the average sequence, the second part of this article focuses on the scan of the space of compact multiple alignments in order to provide an average sequence of a set of sequences. We propose to use a genetic algorithm based on a specific representation of the genotype inspired by genes. This representation of the genotype makes it possible to consistently paint the fitness landscape.

Experiments carried out on standard datasets show that the proposed approach outperforms existing methods.

Key words: Time Series Summarizing, Time Series Averaging, Dynamic Time Warping, Multiple Alignment, Compact Multiple Alignment, Consensus Sequence, Soft Computing.

1. Introduction

Summarizing a set of sequences is an old topic that has been revived in the last decade, due to the increasing availability of sequential datasets. The definition of a consensus object is on the center of data mining issues, since it crystallizes the underlying organization of the data. When dealing with time series or with sequences, the temporal aspect of the data has to be taken into account in this summarization. Finding a consensus (summary) representation of a set of sequences is even described by Dan Gusfield as the Holy Grail [1], through the concept of multiple alignment. In addition, this summary relies on the measure used to compare the data *i.e.*, the sequences. In a naive way, if the length of the sequences was the only point of view, a consensus would be any sequence with a length equal to the average length of the sequences. Obviously, it is an unrealistic example, but it points out that the formulation of a consensus is linked to the meaning given (by the analyst) to the similarity between sequences.

The *Dynamic Time Warping* similarity measure (DTW, for short) was introduced in [2] with applications in speech recognition. DTW is currently a well-known similarity measure on time series (or numerical

*Corresponding author – LSIIT, Pôle API, Bd Sébastien Brant, BP 10413, 67412 Illkirch Cedex, France – Tel.: +33 3 68 85 45 78 – Fax.: +33 3 68 85 44 55

Email addresses: fpetitjean@unistra.fr (François Petitjean), gancarski@unistra.fr (Pierre Gançarski)

Article accepted for publication in Theoretical Computer Science accessible at doi:10.1016/j.tcs.2011.09.029

sequences), and is widely recognized as a relevant measure in various applications [3–8]. The keyword in the last sentence is the word "relevant"; since it concentrates the meaning given to the data. Actually, many consensus sequences can be built in order to summarize a set of sequences, depending on the desired information to be extracted from the sequences. DTW is considered to be a sound measure to understand the hidden structure of temporal datasets, since it is able to capture distortions in the temporal axis. Hence, assuming that DTW is relevant to compare time series, the definition of a consensus sequence of a dataset has to be adapted, according to the behavior of DTW.

If consensus representations are easily definable for objects in the Euclidean space, this is much more difficult for sequences compared with DTW. In the context of time warping, the term "consensus" generally covers three meanings: (1) the longest common subsequence of a set, (2) the medoid sequence of a set, and (3) the average sequence of the set. The longest common subsequence generally permits to visualize a summary of a set of sequences, but its use is however very limited, since the common subsequence does not cover the whole summarized set of sequences. The two other concepts refer to a more classic definition, corresponding to the sequence in the *center* of the set of sequences. The *center* notion has then to be detailed. The commonly accepted definition is the sequence minimizing the sum of (squared) distances to the sequences of the dataset (Definition 1). When the center must be a sequence of the dataset, the center is called "medoid sequence". Otherwise, when the search space of the center is not restricted, the most widely used term is "average sequence".

Definition 1. *Let E be the space of the coordinates of sequences. By a minor abuse of notation, E^T is used to designate the space of all sequences of length T. Given a set of sequences $\mathcal{S} = \{S_1, \cdots, S_N\}$, the average sequence $\mathcal{C}$, consistent with DTW, has to fulfill:*

$$\forall X \in E^T, \sum_{n=1}^{N} DTW^2(\mathcal{C}, S_n) \leqslant \sum_{n=1}^{N} DTW^2(X, S_n) \tag{1}$$

The medoid notion adds a constraint on the space of $\mathcal{C}$ with $\mathcal{C} \in \mathcal{S}$.

Note that this sum, is often called *Within Group Sum of Squares* (WGSS), *discrepancy distance* in [9], or *inertia* in data mining. In addition, the definition of the consensus sequence relies on the *Steiner* trees theory[1], since the average sequence of Definition 1 is named Steiner sequence in this theory.

As recalled in [11], when objects of interest are simple points in an Euclidean space, the minimization problem corresponding to Equation (1) can be solved by using the property of the arithmetic mean. This article details the solution to the average sequence problem under time warping, since the notion of the arithmetic mean is not easily extensible to semi-pseudometric spaces (*i.e.*, spaces induced by semi-pseudometrics like DTW).

The need for an averaging method suitable to DTW is illustrated by numerous papers, either depicting the need for an averaging method [12–14], or proposing heuristics of averaging [9, 15, 16]. All of these methods are averaging the sequences pairwise, leading to inaccurate average sequences, since these methods are non-associative, with no guarantee that a different order would lead to the same result. In order to solve this problem, we recently introduced in [17] an optimization method named Dtw Barycenter Averaging (DBA). This method consists in refining a given average sequence in order to make it converge to a (generally local) minimum of the inertia. DBA was showed to outperform all other heuristics to the minimization problem.

This article is divided into two main parts. The first part describes the theoretical contributions, while the second part shows how these contributions can be used to average a set of time series.

The first part starts by recalling the definition of DTW in Section 2. Then, Section 3 introduces the notion of *compact multiple alignments* and their usefulness for averaging a set of sequences. We show that averaging under time warping is directly linked to the notion of multiple alignment and that the new concept

[1] The Steiner problem consists in finding a shortest network connecting all the points of a set [10]. Note that most Steiner problems are NP-complete.

of compact multiple alignment is required to derive a synthetic representation of a set of sequences. Finally, this section gives a representation of compact multiple alignments and provides some properties.

The second part starts by presenting an evolutionary strategy in Section 4, covering the space of compact multiple alignments towards the average sequence. This strategy relies on a specific representation of the genotype. Experiments carried out on standard datasets from the UCR time series classification and clustering archive [18] are conducted in Section 5 in order to compare our method to existing ones. Finally, Section 6 concludes the article and presents some further works.

2. Dynamic Time Warping (DTW)

This section recalls the definition of the Euclidean distance and of the DTW similarity measure. Throughout this section, let $A = \langle a_1, \dots, a_T \rangle$ and $B = \langle b_1, \dots, b_T \rangle$ be two sequences, and let δ be a distance between elements (or *coordinates*) of sequences, *e.g.*, the Euclidean distance.

2.1. Euclidean distance between sequences

This distance is commonly accepted as the simplest distance between sequences. The distance between the two sequences A and B is defined by:

$$D(A, B) = \sqrt{\delta(a_1, b_1)^2 + \cdots + \delta(a_T, b_T)^2} \tag{2}$$

Unfortunately, this distance does not correspond to the common understanding of what a sequence really is, and cannot capture flexible similarities. For example, the two sequences $\langle x, y, x, x \rangle$ and $\langle x, x, y, x \rangle$ are different according to this distance even though they represent similar trajectories in the coordinate space.

2.2. Dynamic Time Warping

DTW is based on the Levenshtein distance (also called edit distance) and was introduced in [2, 19], with applications to speech recognition. It finds the optimal alignment (or coupling) between two sequences of numerical values, and captures flexible similarities by aligning the elements inside both sequences. The cost of the optimal alignment can be recursively computed by:

$$D(A_i, B_j) = \delta(a_i, b_j) + \min \left\{ \begin{array}{l} D(A_{i-1}, B_{j-1}) \\ D(A_i, B_{j-1}) \\ D(A_{i-1}, B_j) \end{array} \right\} \tag{3}$$

where A_i (resp. B_j) denotes here the subsequence $\langle a_1, \dots, a_i \rangle$ (resp. $\langle b_1, \dots, b_i \rangle$). The overall similarity is given by $D(A_{|A|}, B_{|B|}) = D(A_T, B_T)$.

Unfortunately, a direct implementation of this recursive definition leads to an algorithm that has exponential cost in time. Fortunately, the fact that the overall problem exhibits overlapping sub-problems allows for the memoization of partial results in a matrix, which makes the minimal-weight coupling computation a process that costs $|A| \cdot |B|$ basic operations. This measure has thus a time and a space complexity of $O(|A| \cdot |B|)$.

DTW is able to find optimal global alignment between sequences and is probably the most commonly used measure to quantify the dissimilarity between sequences [4–8]. It also provides an overall real number that quantifies the similarity. An example of an alignment computed by DTW between two sequences can be found in Figure 1: it shows the alignment of points taken from two sinusoids, one being slightly shifted in time. The numerical result computed by DTW is the sum of the heights[2] of the associations. Alignments at both extremities in Figure 1 show that DTW is able to correctly re-align one sequence with the other, a process which, in this case, highlights similarities that the Euclidean distance is unable to capture. Algorithm 1 details the computation.

[2] In fact, the distance $\delta(a_i, b_j)$ computed in Equation 3 is the distance between two coordinates without considering the time distance between them.

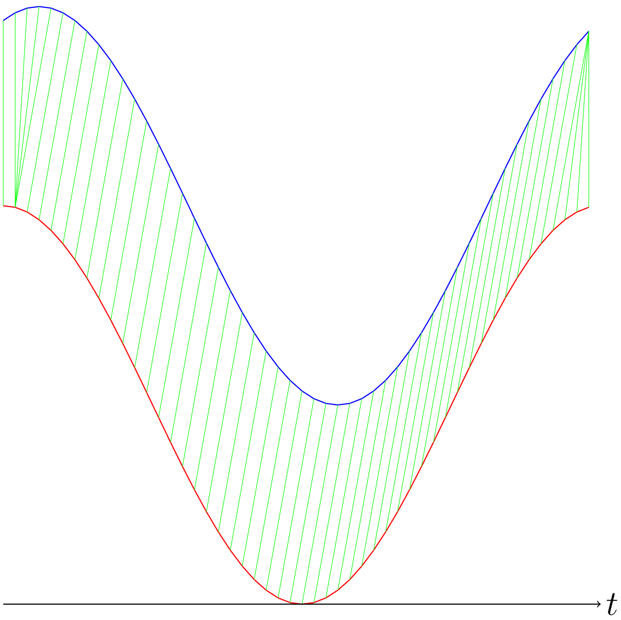


Figure 1: Two 1D sequences aligned with Dynamic Time Warping. Coordinates of the top and bottom sequences have been respectively computed by $\cos(t)$ and $\cos(t+\alpha)$. For visualization purpose, the top sequence is drawn vertically shifted.

Algorithm 1 DTW

```
Require: $A = \langle a_1, \ldots, a_S \rangle$ with $S = |A|$
Require: $B = \langle b_1, \ldots, b_T \rangle$ with $T = |B|$
Let $\delta$ be a distance between elements of sequences
Let $m[S,T]$ be a cost matrix
  $m[1,1] \leftarrow \delta(a_1, b_1)$

  for $i \leftarrow 2$ to $S$ do
    $m[i,1] \leftarrow m[i-1,1] + \delta(a_i, b_1)$
  end for
  for $j \leftarrow 2$ to $T$ do
    $m[1,j] \leftarrow m[1,j-1] + \delta(a_1, b_j)$
  end for

  for $i \leftarrow 2$ to $S$ do
    for $j \leftarrow 2$ to $T$ do
      $m[i,j] \leftarrow \delta(a_i, b_j) + min \begin{cases} m[i-1\,, \quad j \quad] \\ m[\quad i \quad, j-1] \\ m[i-1\,, j-1] \end{cases}$
    end for
  end for
  return $m[S,T]$
```

A	a_1	$-$	$-$	a_2	a_3	a_4	a_5
B	b_1	$-$	b_2	b_3	b_4	b_5	$-$
D	d_1	d_2	d_3	d_4	d_5	$-$	$-$

(a)

A	1	1	1	10	0	0	4
B	0	0	2	10	0	0	0
D	0	0	0	10	0	0	0
$\mathcal{C}$	$\frac{1}{3}$	$\frac{1}{3}$	1	10	0	0	$\frac{4}{3}$

(b)

Figure 2: (a) Multiple alignment of three sequences A, B and D with $A = \langle 1, 10, 0, 0, 4\rangle$, $B = \langle 0, 2, 10, 0, 0\rangle$ and $D = \langle 0, 0, 0, 10, 0\rangle$. The symbols '–' denotes the repetition of the previous state. (b) Numerical values of the sequences as well as the consensus sequence $\mathcal{C}$.

3. Compact Multiple Alignment

In the following, $\mathcal{S} = \{S_1, \cdots, S_N\}$ denotes a set of N sequences from which we want to compute a consensus sequence $\mathcal{C}$, and T denotes the length of the sequences belonging to $\mathcal{S}$.

Section 3.1 gives the intuition on the connection between the Steiner theory, the multiple alignment and the average sequence. Section 3.2 introduces the concept of compact multiple alignment. Finally, Section 3.3 gives a representation of compact multiple alignments as well as several properties.

3.1. Background

The concept of multiple sequence alignment was introduced in [20] for the alignment of N sequences all together. This multiple alignment is computable in generalizing DTW for the alignment of N sequences. For instance, instead of computing DTW by comparing three values in a matrix (Algorithm 1), three sequences are aligned by comparing seven values in a three dimensional matrix (cube). Figure 2(a) illustrates a multiple alignment of three sequences. In the same way, DTW can be computed in a N-dimensional matrix for the comparison/alignment of N sequences. Given this multiple alignment, the consensus sequence can be computed by averaging *column by column* the N aligned sequences (see Figure 2(b) for an example).

This consensus sequence is named *Steiner sequence* in the Steiner theory. It corresponds to the sequence minimizing its distance to other sequences. The formal proof, showing that the multiple alignment specifies the Steiner sequence and conversely, can be found in [1] – Section 14.7.2.

Unfortunately, the multiple alignment process takes $\Theta\left(T^N\right)$ operations [21], and is thus not tractable for more than a few sequences. In addition, the size of the N-dimensional matrix will also be in $\Theta\left(T^N\right)$ which requires unrealistic amounts of memory. Moreover, 30 years of well-motivated research did not provide any exact scalable algorithm, neither for the consensus sequence problem, nor for the multiple alignment problem (see [22–27] for examples of heuristics).

3.2. From multiple alignment to sequence averaging

This subsection aims at showing that a multiple alignment provides an average sequence and conversely. We use the formalism described in [1].

Definition 2. *Let $\mathcal{M}$ be a multiple alignment of the set of sequences $\mathcal{S}$, computed in an N-dimensional matrix (N-matrix, for short). The path linking the connected elements of the N-matrix and providing the multiple alignment $\mathcal{M}$ is named the* warping path *$\mathcal{W}$.*

Definition 3. *Let L be the length of $\mathcal{W}$. $\mathcal{W}$ is a series of 8-connected elements $\langle w_1, \cdots, w_L\rangle$ in the N-matrix where each element w_ℓ is determined by its N coordinates.*

Property 1. *The first and the last elements of $\mathcal{W}$ are respectively $w_1 = \underbrace{(1, \cdots, 1)}_{N}$ and $w_L = \underbrace{(T, \cdots, T)}_{N}$.*

Property 2. *Let us use the function notation for tuples, i.e., let us use $\mathcal{E}(i)$ to denote the i^{th} element of a tuple $\mathcal{E} = (e_1, \cdots, e_n)$. As a direct consequence of Property 1, L has to fulfill:*

$$\begin{array}{ccccc} d_\infty(w_1, w_L) & \leqslant L-1 \leqslant & d_1(w_1, w_L) \\ \max_{n=1}^{N} |w_L(n) - w_1(n)| & \leqslant L-1 \leqslant & \sum_{n=1}^{N} |w_L(n) - w_1(n)| \\ \max_{n=1}^{N} (T-1) & \leqslant L-1 \leqslant & \sum_{n=1}^{N} (T-1) \\ T-1 & \leqslant L-1 \leqslant & N \cdot (T-1) \\ T & \leqslant \quad L \quad \leqslant & N \cdot (T-1) + 1 \end{array} \tag{4}$$

where d_1 and d_∞ respectively denote the Manhattan and the Chebyshev distances.

In summary, $\mathcal{W}$ is a series of connected elements $\langle w_1, \cdots, w_L \rangle$ in the N-matrix where each element w_ℓ is determined by its coordinates in the N-matrix. Thus, each element w_ℓ of the path is associated to one element of each sequence of $\mathcal{S}$, *i.e.*, to N elements.

Definition 4. *Let $\mathcal{W} = \langle w_1, \cdots, w_L \rangle$ be the warping path corresponding the a multiple alignment $\mathcal{M}$ of a set of sequences $\mathcal{S}$. Let λ be the arithmetic mean defined as:*

$$\begin{array}{cccc} \lambda : & \mathbb{R}^n & \to & \mathbb{R} \\ & (v_1, \cdots, v_n) & \mapsto & \frac{1}{N} \cdot \sum_{i=1}^{n} v_i \end{array} \tag{5}$$

Then, the average sequence $\mathcal{C}$ is defined as an average column by column:

$$\begin{array}{lll} \mathcal{C} = & \langle & \\ & & \lambda(S_1(w_1(1)), \cdots, S_N(w_1(N))) \quad , \\ & & \vdots \\ & , & \lambda(S_1(w_L(1)), \cdots, S_N(w_L(N))) \\ & \rangle & \end{array} \tag{6}$$

Definition 5. *Let $\mathcal{M}_\mathcal{C}$ be the multiple alignment from which the derived average sequence is optimal, i.e., fulfilling Equation 1. $\mathcal{M}_\mathcal{C}$ is associated to its warping path $\mathcal{W}_\mathcal{C}$ of length $L_\mathcal{C}$.*

As a consequence of Definition 4, the length of the average sequence is $L \in [1\ ,\ N \cdot (T-1) + 1]$. Realistically, as N is generally in $\left[10^2, 10^6\right]$, $\mathcal{C}$ could be thousand times longer than the sequences to be averaged. As this paper focuses on giving a synthetic view of a set of sequences, such a sequence length cannot be seriously considered for an average sequence. In the worst case, this summary would be almost as long as the concatenation of all sequences of the set.

In order to solve this issue, we propose to allow several elements of sequences in each column of the multiple alignment. Intuitively, it corresponds to a contraction of the width of $\mathcal{M}_\mathcal{C}$. In fact, the aim is to obtain a path $\mathcal{W}$ with a given length L', close to the length of sequences of $\mathcal{S}$, *i.e.*, with $L' \approx T$. However, such a path would lead to an Euclidean alignment, where every i^{th} element of the sequences "belongs" to the same column i of $\mathcal{M}$. Hence, we propose to model a multiple alignment $\mathcal{M}$ by a multiple alignment with fewer columns but where each column can contain several successive elements of each sequence.

Definition 6. *Let $\mathcal{M}_\mathcal{C}$ be the optimal multiple alignment associated to its warping path $\mathcal{W}_\mathcal{C} = \langle w_1, \cdots, w_L \rangle$ of length $L_\mathcal{C}$. $\mathcal{M}_\mathcal{C}^\star$ denotes the compact multiple alignment of $\mathcal{M}_\mathcal{C}$, associated to its warping path $\mathcal{W}_\mathcal{C}^\star = \langle w_1^\star, \cdots, w_L^\star \rangle$ of length $L^\star$ with:*

$$L^\star \ \approx\ T \ \ll\ L_\mathcal{C}^{\max} = N \cdot (T-1) + 1 \tag{7}$$

A	$\{a_1\}$	$\{a_2\}$	$\{a_3, a_4, a_5\}$
B	$\{b_1, b_2\}$	$\{b_3\}$	$\{b_4, b_5\}$
D	$\{d_1, d_2, d_3\}$	$\{d_4\}$	$\{d_5\}$

(a)

A	$\{1\}$	$\{10\}$	$\{0, 0, 4\}$
B	$\{0, 2\}$	$\{10\}$	$\{0, 0\}$
D	$\{0, 0, 0\}$	$\{10\}$	$\{0\}$
$\mathcal{C}$	$\frac{1}{2}$	10	$\frac{2}{3}$

(b)

Figure 3: (a) Compact multiple alignment of three sequences A, B and D with $A = \langle 1, 10, 0, 0, 4 \rangle$, $B = \langle 0, 2, 10, 0, 0 \rangle$ and $D = \langle 0, 0, 0, 10, 0 \rangle$. (b) Numerical values of the sequences as well as the consensus sequence $\mathcal{C}$.

Then, in order to fulfill Equations 4 and 7, every $w_\ell^\star$ is defined as a set of sets of elements of sequences, *i.e.*, instead of being linked to one element of each sequence of $\mathcal{S}$, every $w_\ell^\star$ is linked to a set of successive elements of each sequence of $\mathcal{S}$. In this way, all previous definitions remain valid by replacing $\mathcal{W}_\mathcal{C}$ with $\mathcal{W}_\mathcal{C}^\star$. A compact multiple alignment with $L^\star = 3$ is illustrated in Figure 3(a).

Furthermore, giving an average sequence $\mathcal{C}$, a compact multiple alignment $\mathcal{M}_\mathcal{C}^\star$ can be computed using DTW. Actually, DTW is able to provide an alignment between $\mathcal{C}$ and each sequence of $\mathcal{S}$, resulting in a coupling between each element of $\mathcal{C}$ and several elements of the sequences of $\mathcal{S}$. Note that this description also fits the definition of a multiple alignment.

Definition 7. *Let $\mathcal{M}^\star$ be a compact multiple alignment, let $\mathcal{C}$ be the resulting average sequence and let $\mathcal{M}^{\star+}$ be the compact multiple alignment computed from $\mathcal{C}$. The function computing $\mathcal{C}$ from $\mathcal{M}^\star$ is named f while the function computing $\mathcal{M}^{\star+}$ from $\mathcal{C}$ is named g. The function f is defined by the Equation 6 where $\mathcal{W}$ is replaced by $\mathcal{W}^\star$. The function g is defined by the computation of DTW between $\mathcal{C}$ and and every sequence of $\mathcal{S}$.*

Property 3. *f is not the inverse function of g and conversely.*

Proof. *Let $\mathcal{M}_1^\star$ be a compact multiple alignment of one sequence $S = \langle 1, 2, 2 \rangle$, defined as $\{\{1, 2\}, \{2\}\}$, i.e., the two first coordinates of S are in the first column of $\mathcal{M}_1^\star$ and the last coordinate of S is in the last column of $\mathcal{M}_1^\star$. We have then the average sequence $\mathcal{C}_1 = f(\mathcal{M}_1^\star) = \langle 1.5\ ,\ 2 \rangle$. Then, the resulting multiple alignment $\mathcal{M}_1^{\star+}$ is defined as $\mathcal{M}_1^{\star+} = g(\mathcal{C}_1) = \{\{1\}, \{2, 2\}\}$ thanks to DTW. Hence, $g \circ f(\mathcal{M}_1^\star) \neq \mathcal{M}_1^{\star+}$.*

Furthermore, the $g \circ f$ function can be used as an optimization method. This method was introduced in [17] out of the scope of multiple alignment; it consists in applying several times the $g \circ f$ function from a sequence of the dataset.

3.3. Properties of a compact multiple alignment

Last subsection described the construction of the average sequence from the multiple sequence alignment. Abusing the notation, we use "multiple alignment" in the sequel of this article to denote a "compact multiple alignment". For visualization purpose, and without loss of generality, this subsection illustrates the representation of a multiple alignment on a single sequence.

We choose to represent a multiple alignment by a cover of the elements of the sequence, for each sequence. Thus, the multiple alignment is represented by a list of N covers; one cover by sequence of $\mathcal{S}$. Figure 4 shows the representation of a multiple alignment on the left, and its corresponding average sequence computed with the function f on the right. The two representations are identical.

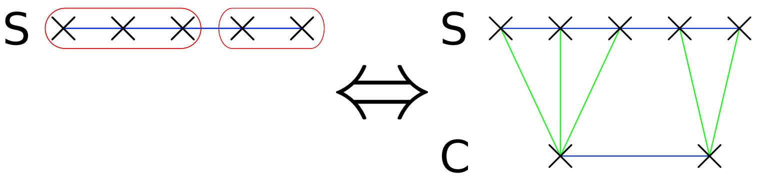


Figure 4: On the left, the multiple alignment of a single sequence S with two groups of coordinates is illustrated, *i.e.*, corresponding to a warping path of length two. On the right, the derived average sequence C is depicted. The first set of elements means that the three first elements of S are in the first column of the compact alignment, *i.e.*, linked to the first element of C. The second set of elements means that the two last elements of S are in the last column of the compact multiple alignment, *i.e.*, linked to the second element of C.

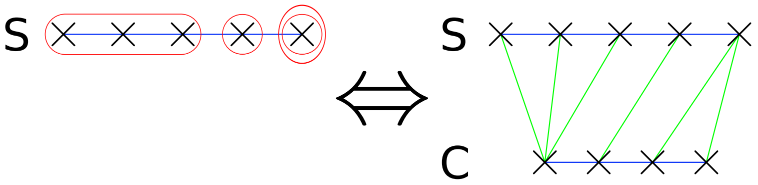


Figure 5: The first set of elements means that the three first elements of S are in the first column of the compact multiple alignment, *i.e.*, linked to the first element of C. The second set of elements means that the fourth element of S is in the second column of the multiple alignment, *i.e.*, linked to the second element of C. The third set (any of the two overlapping sets) means that the last coordinate of S is in the third column of the multiple alignment, *i.e.*, linked to the third element of C. Finally, the last set means that the last element of S is also in the last column of the multiple alignment, *i.e.*, linked to the last coordinate of C.

This part will give some properties of the representation of a multiple alignment by a list of set of sets of sequence elements (one set of sets of elements by sequence in the set). These properties allow us to restrict the space of potential solutions.

Property 4. *There are as many sets in each cover as there are elements in the average sequence.*

Proof. By definition of the given representation, all ℓ^{th} sets of elements of the sequences to be averaged, correspond to the ℓ^{th} column of the multiple alignment. By definition of the average sequence of a multiple alignment, the ℓ^{th} element of the average sequence is defined as the barycenter (computed with the arithmetic mean) of elements from all ℓ^{th} sets of the multiple alignment.

Property 5. *Each element of the sequences of $\mathcal{S}$ must be in a column of the multiple alignment, i.e., must be part of a set.*

Proof. By definition, DTW cannot skip any element of the sequences, so cannot the average sequence neither.

Property 6. *An element of a sequence can be part of several columns, i.e., can be linked to several elements of the average sequence. Thus, the sets of elements of a sequence are not forming a partition but a cover, making these sets overlap (illustrated in Figure 5).*

Proof. Since several elements of a sequence can be part of one column of the multiple alignment (Definition 6), as well as each element must be part of a column (Property 5), an element of a sequence can be part of several columns.

Property 7. *The overlap cannot cover more than one element of a sequence.*

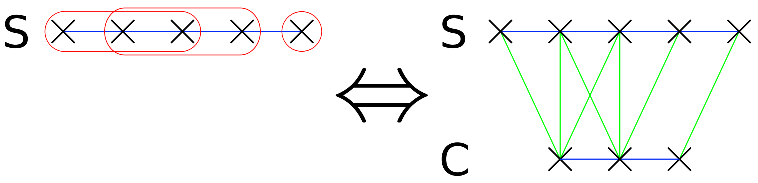


Figure 6: Impossible coupling of DTW. The first set of elements means that the three first elements of S are linked to the first element of C. The second set means that the second, the third and the fourth elements of S are linked to the second element of C. The last set means that the last element of S is linked to the last element of C. This coupling is not correct since the links are crossing each other.

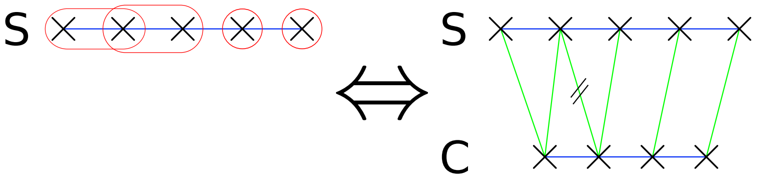


Figure 7: Impossible coupling of DTW. The first set of elements means that the two first elements of S are linked to the first element of C. The second set means that the second and the third elements of S are linked to the second element of C. The third (resp. fourth) set means that the fourth (resp. fifth) element of S is linked to the third (resp. fourth) element of C.

Proof. If this overlap was covering more than one element, it would lead to a coupling where the links are crossing each other (illustrated in Figure 6).

Property 8. *The overlap can cover a single element of a sequence.*

Proof. The proof of the existence of such a case is illustrated in Figure 5.

Property 9. *Only singletons can overlap.*

Proof. The proof of the existence and validity of the overlap of singletons is illustrated in Figure 5. If a set with a cardinality superior to one was overlapping, the coupling would comprise an "N" pattern (illustrated in Figure 7), which is not a possible coupling for DTW, since the coupling without the middle link (crossed out in Figure 7) of this "N" pattern is a better coupling.[3]

Conclusion

This section introduced the concept of compact multiple alignment. Compact multiple alignment makes it possible to build a concise average sequence of a set of sequences. We showed that a compact multiple alignment of N sequences can be represented by N covers of elements of the N sequences. This representation of the space of the solutions makes the number of solutions countable. Thus, even though the original space of the average sequence is uncountable ($\mathbb{R}^{L^\star}$), the space of the multiple alignments is countable and makes it possible to efficiently scan the solution space.

The next section will present how a genetic algorithm can be used to scan the space of compact multiple alignments, in order to find an average sequence with respect to the criteria given in Equation 1.

[3] In fact, if the two elements linked by the middle link of the "N" were at a distance of zero, this coupling would remain valid. However, it would not be better than those without the middle link. This pattern is thus forbidden, since it reduces the solution space, while providing as good results as those allowing this pattern.

4. A genetic averaging method for DTW

Genetic algorithms (GAs) are known to be well-suited both for Steiner problems [28] and for multiple alignments [22]. The main evolution theories are recalled in Appendix A. This section presents the use of GAs for the search of the best compact multiple alignment, in order to build an average sequence of a set of time series.

The proposed approach includes a specific model of the genome based on compact multiple alignments (*i.e.*, the potential average sequence), and a local optimization of the solution (Lamarckian approach). The genotype represents the compact multiple alignment while the phenotype is the average sequence, obtained by applying the f function to the genotype.

The objective function (corresponding to the evaluation of phenotypes) is the minimization of the sum of the squared distances from the phenotype to sequences of the dataset to summarize. The proposed approach is named COmpact Multiple Alignment for Sequence Averaging (COMASA, for short).

COMASA processes as a standard genetic procedure. It starts with an initial set of genotypes (compact multiple alignments) and iterates as follows:

1. Compute phenotypes (average sequences) from genotypes with a local optimization process;
2. Evaluate every phenotype with Equation 1;
3. Keep a mix of new and old solutions;
4. Build of a new set of solutions from previous solutions (with crossovers and mutations) and possibly adding new random solutions.
5. Iterate to step 1.

This section details: (1) the genotype representation and its initialization; (2) the crossover and the mutation functions; (3) the local optimization process and (4) the evaluation function. Other experimental details, *i.e.*, the evolution strategy used as well as all weights and parameters are given in the next section.

4.1. Genotype – Phenotype

4.1.1. Model – data structure

A good genotype has to consistently represent the fitness landscape while being easily manipulable and taking few memory space. The aim of a good representation is to find the right balance between these three characteristics. In our case, the most important is the representation of the fitness landscape since the solution to the problem we are handling requires $\Theta\left(T^N\right)$ operations. Therefore the genotype was designed in order to isolate the influence of each sequence on the solution.

Section 3 showed that a compact multiple alignment can be represented by as many covers as there are sequences to average, *i.e.*, N. These respective N covers of the N sequences are independent from each other. The constraints on these covers have been given in Section 3.3. In this way, providing a cover of the elements of each sequence of the dataset to summarize, the phenotype (average sequence) is provided by the function f. A genotype is thus designed as a list of N covers. In parallel with biological genomes, each one of the N covers is called a gene. For instance, the n^{th} gene of this list will represent the dispatching of the elements of the n^{th} sequence to the elements of the average sequence.

In practice, each cover can be implemented as a table of size T associating each element of a sequence to the set of columns in which the element belongs (see Figure 8). A genotype can be thus represented as a $N \times T$-matrix of sets.

$\{1\}$	$\{1\}$	$\{1\}$	$\{2\}$	$\{3,4\}$

Figure 8: Representation of one gene (cover) of the genotype. This example represents the configuration given in Figure 5.

4.1.2. Random initialization of the genotype

Let L be the length of the average sequence, *i.e.*, the number of columns of the compact multiple alignment, as a parameter of the method. Giving this representation of the genotype, the building of a random genotype is not as trivial as if the genotype was modeled by the coordinates of the average sequence. N random covers have to be generated, corresponding to the N sequences. Moreover, each cover has to satisfy all the properties expounded in the Section 3.3. These N initialization are hopefully independent. Thus, the initialization of the genotype focuses on the generation of a random cover of the elements of a sequence. This cover is shaped as a partition of theses elements, with the possibility to replicate singletons in order to form L sets (as many sets as the number of elements of the average sequence). Starting with a single set containing all elements of the n^{th} sequence, this set is then randomly split $(L-1)$ times. Splitting a singleton duplicates this set in the cover. Every gene need L operations to be initialized. Thus, the initialization of a genotype takes $\Theta(N \cdot L)$ operations.

4.2. Reproduction functions

4.2.1. Crossover function

Every genotype comprises N genes and the child genotype must also be composed of N genes. The genes have to remain atomic, since the crossover of two covers is not simple to define, especially when the resulting cover has to fulfill the properties 5 to 9. Thus, the n^{th} gene of the child genotype is chosen among the n^{th} genes of each one of the two parents. This allows us to make the child genotype consistent with the properties of Section 3.3. In the same way than two genomes are crossed in biology, two genotypes are crossed by taking randomly either a gene from the first genotype, or a gene from the second genotype.

This process is efficient since it enables the crossover of the elements *coding* for a phenotype, and not directly the crossover of the phenotype. This generation of a new genotype from two parents takes $\Theta(N)$ operations.[4]

4.2.2. Mutation function

The mutation function prevents from a premature convergence of the process in introducing diversity in the solutions. Therefore, the mutation operator consists in slightly modifying the covers of a genotype.

The proposed mutation operator consists in perturbing a gene by merging two successive sets of the cover and to split one set of the resulting cover. Therefore an element in the sequence is randomly chosen, there are then two cases: (1) the element belongs to several sets of the cover, then one of this set is deleted from the cover; (2) the element belongs to a single set of the cover, then this set is randomly merged with its left or right neighbor.

After this reduction of the number of sets in a cover, one set of the resulting cover is split following the same scheme as for gene generation. Thus, the mutation of one gene of the genotype takes $\Theta(1)$ operations.

4.3. Evaluation

The evaluation function corresponds to the initial objective given for the average sequence, namely the sum of the squares, also known as inertia, given in Equation 1.

Even if the computation of the evaluation function for a solution is trivial, one has to note that evaluating one solution consists in computing DTW between itself and every sequence of $\mathcal{S}$. Therefore DTW has to be computed once per sequence to average, that is N times. The complexity of DTW is $\Theta\left(T^2\right)$ *i.e.*, $\Theta\left(L \cdot T\right)$ in our case. The complexity of the evaluation of one solution is therefore $\Theta\left(N \cdot L \cdot T\right)$.

[4] Suppose that a genotype is directly represented as a series of sequence elements (our phenotype). The crossover of two such genotypes would lead to a new genotype which is not composed of barycenters of coordinates. However, every coordinate of the optimal average sequence is compulsorily a barycenter of coordinates of sequences to summarize. In this way, the data structure of the genotype makes it possible to provide a relevant crossover operation which significantly improves the convergence of the process.

4.4. Local optimization process

In order to speed up the convergence of the process, our framework includes a local optimization process. Before the evaluation of individuals, the solutions are locally optimized by the DBA process introduced in [17]. As shown in Section 3.2, this local optimization corresponds to the application of $g \circ f$ before the evaluation. However, this framework is not limited to this optimization method and another process could be used.

The f and g functions respectively require $\Theta(N \cdot T)$ and $\Theta(N \cdot L \cdot T)$ operations. Thus, the evaluation of the genotypes is made using $f \circ g \circ f$ applied on the genotype (*i.e.*, the compact multiple alignment).

4.5. Complexity

This section details the complexity of the algorithm. Let N be the number of sequences to average, T be the mean length of the sequences to average[5], L be the length of the average sequence, P be the size of the population, G be the number of generations.[6] Table 1 summarizes the complexity of COMASA for one generation.

Table 1: Complexity of COMASA for one generation.

Step	**Time complexity**
Initialization	$\Theta(P \cdot N \cdot L)$
Evaluation	$\Theta(P \cdot N \cdot L \cdot T)$
Crossover	$\Theta(P \cdot N)$
Mutation	$\Theta(P \cdot N)$

Finally, the COMASA processes overall in:

$$\begin{array}{rc} & \Theta(G \cdot (P \cdot N \cdot L + P \cdot N \cdot L \cdot T + 2 \cdot P \cdot N)) \\ = & \Theta(G \cdot P \cdot N \cdot (L + L \cdot T + 1)) \\ = & \Theta(G \cdot P \cdot N \cdot L \cdot T) \end{array} \tag{8}$$

5. Experiments

This section aims at comparing COMASA to existing methods. Consequently, COMASA is compared to: (1) NLAAF, which was introduced in [15] and consists in an progressive averaging of a set of sequences with a tournament scheme; (2) DBA, which was already presented in this article. The average sequences are evaluated on their ability to minimize the sum of Equation 1. All experiments are carried out on standard time series datasets [18]. Figure 9 shows a sequence per class, for each dataset in the archive.

To make these experiments reproducible, we detail here the experimental settings:

- all programs are implemented in Java and run on an Intel® Core™ 2 Quad processor running at 2.8 GHz with 8 GB of RAM;
- the distance used between two elements of sequences is the squared Euclidean distance. As the square function is a strictly increasing function on positive numbers, and since only comparisons between distances are used, it is unnecessary to compute square roots. The same optimization has been used in [29], and is rather common;

[5] Note that our approach makes it possible to average sequences with various lengths.

[6] The ratio of kept individuals from one generation to the next, as well as the probability of mutation do not modify the complexity of the process since they are multiplicative factors in $[0, 1]$. Thus, we consider that they are set to one.

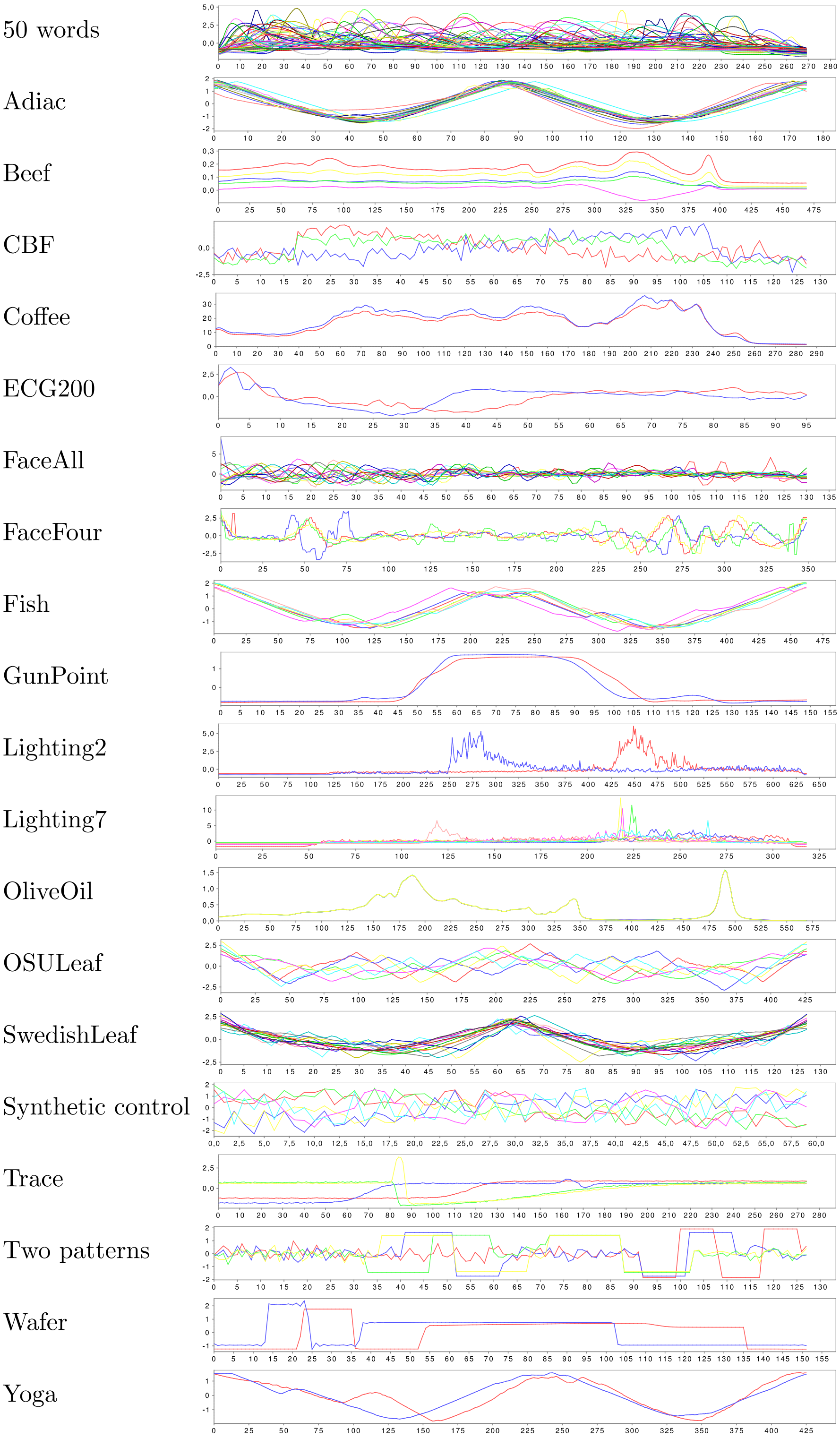


Figure 9: Sample time series extracted from the datasets of the archive used. One time series from each class is displayed for each dataset.

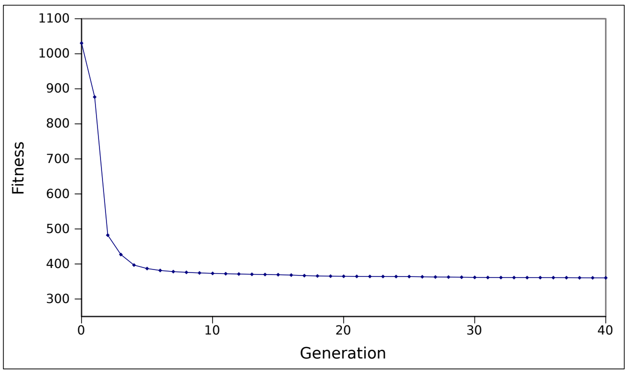


Figure 10: Convergence of the process over 40 generations on the *Yoga* dataset. This dataset is one of the "hardest" since it is composed of $3,300$ sequences of lengths 426.

- sequences have been normalized with Z-score: for each sequence, the mean $\bar{x}$ and standard deviation σ of the coordinate values are computed, and each coordinate y_i is replaced by:

$$y'_i = \frac{y_i - \bar{x}}{\sigma} \tag{9}$$

- as the aim is to test the capacity of COMASA to minimize the inertia, and because the focus is not on supervised methods, all sequences from both *train* and *test* dataset are put together ;
- an average sequence per class of each dataset is computed.

Moreover, any genetic algorithm has some computing parameters. The different parameters are let as general as possible in order to show the relevance of the process. The algorithm was parametrized as follows:

- the length of the average sequences is the same as the one of the dataset ($L = T$), since the aim of this summarizing framework is to obtain a synthetic representation of the time series. Moreover this length is shown to be sound, according to [17][7];
- one percent of the genes of every genotype is mutated (one merge and one split); this weighting is rather common;
- one iteration of DBA ($g \circ f$) is performed when computing the phenotype from the genotype;
- the population size is fixed to 100 for memory purposes;
- the number of generations is fixed to 50 (Figure 10 shows that the process has generally converged afterwards);
- we use the tournament scheme for the selection of individuals;
- ten percent of new individuals are introduced at each generation.

In order to give some qualitative overview, Figure 11 gives an example of the behavior of COMASA on synthetic time series while Figure 12 illustrates two average sequences computed by COMASA on two sets of sequences. It can be noticed in Figures 12(c) and 12(d) that the average sequence of one class of the *50words* dataset doest not look like any sequence of the class. Actually, in the space of sequences, the average sequence can be different from the sequences it averages. This phenomenon is very same in an Euclidean

[7] In an intuitively way, one can see that the sum of the squares of the average sequence in Figure 2(b) is 14, while the one of the average sequence in Figure 3(b) is about 16.1, whereas the second one is more than two times shorter.

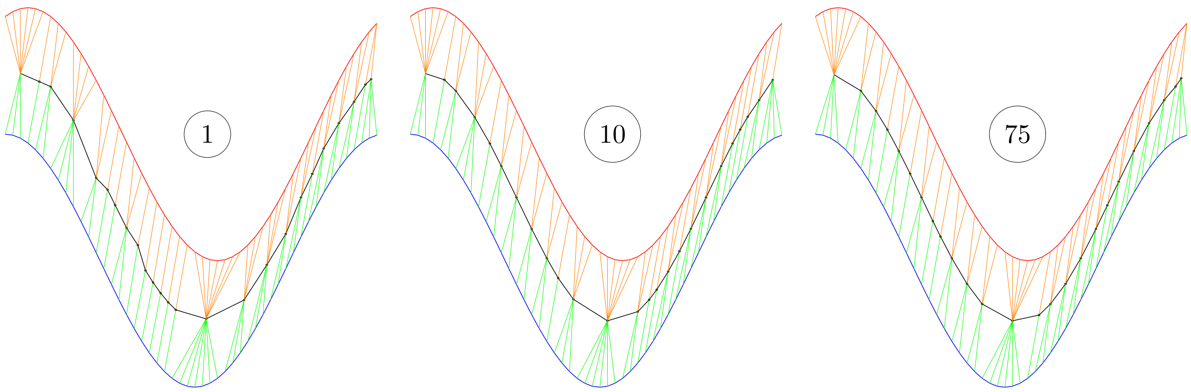


Figure 11: Example of the behavior of COMASA on generations one, ten and 75. For visualization purposes, no local optimization was used in order to slow down the process. The first solution shows that the data structure of the genotype limits the solutions space and provides at the first generation, an already significant solution. This first solution provided is actually far from a random sequence.

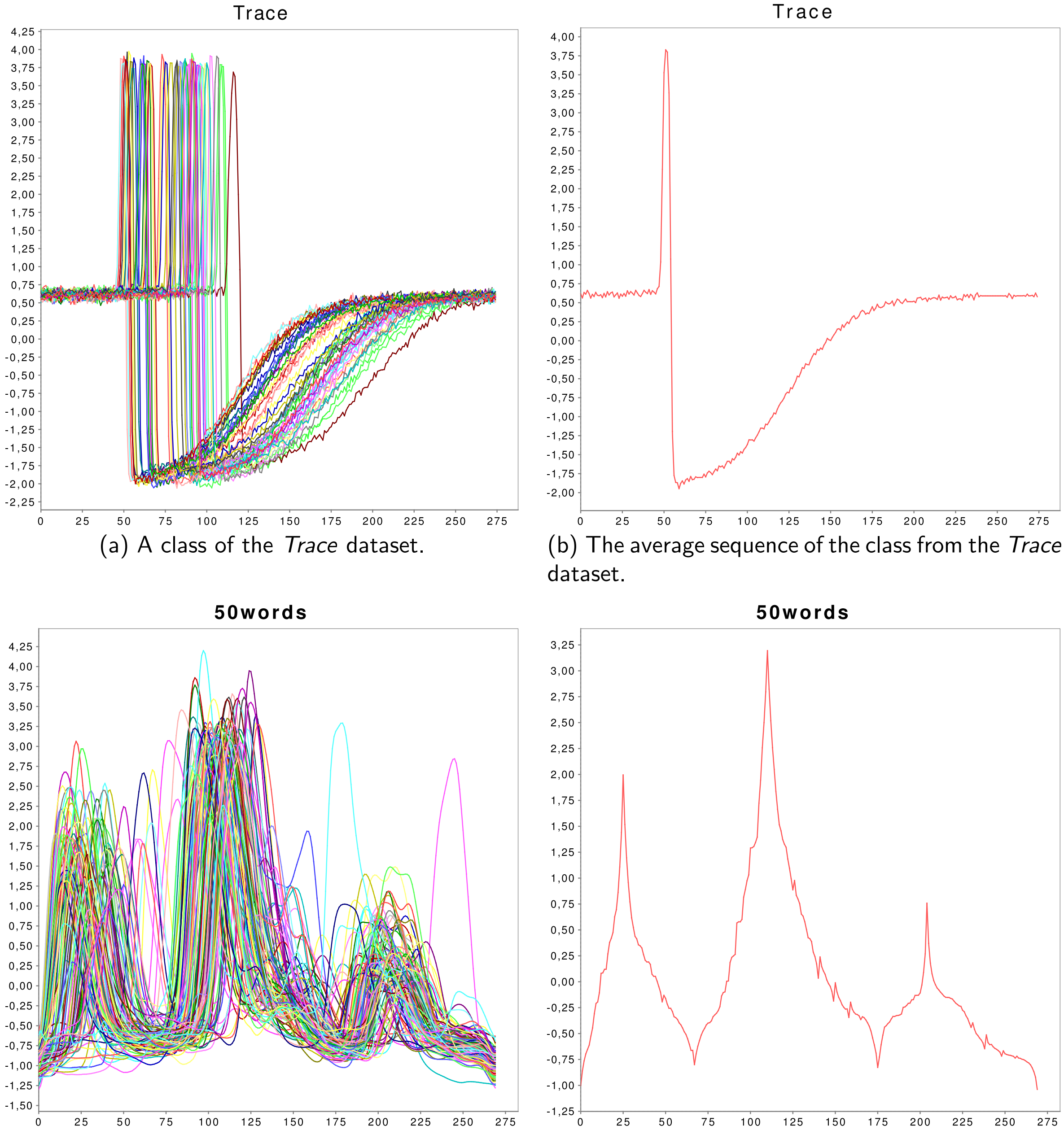
space where the barycenter of a set of points can be quite far from them. Moreover, the sharp shape of the average sequence in Figure 12(d) is quite interesting. We have seen in Equation 4 that the length L of the average sequence could be up to $N \cdot (T-1)+1$. In this way, limiting L to T in these experiments, the elements of the average sequence have to be placed where the distance to the elements of the sequences is maximum. Thus, where the derivative of the sequences is about zero, only a few elements are necessary. This phenomenon is clearly visible in Figure 11 where only one element of the average sequence is associated to many elements of the two averaged sequences, in the valley part of the sinusoids.

Table 2 shows the global inertia obtained for each dataset. First of all, the scores obtained by the medoid confirm that, contrary to the intuition, even if the medoid is similar in shape to the averaged sequence, it cannot be used in place of the average sequence.

Secondly, one can notice that, for all datasets, COMASA reduces/improves the intraclass inertia. COMASA provides average sequences with scores 47 % lower than the scores of NLAAF and 20 % lower than the scores of DBA which is a significant improvement.[8] Such improvements show that the choice of the averaging method has to be seriously considered if the averaging method is an important step (database indexing, machine learning, etc.). Indeed, such improvements point out that the average sequences used were not exactly at the center of the classes, which could disrupt the functioning of a broader process. These experiments demonstrates both the relevance of compact multiple alignments to summarize a set of times series, and the adequacy of evolutionary frameworks to solve multiple alignments problems and correlated ones.

Table 2 gives the runtimes of COMASA, compared to runtimes of DBA. As it was expected, COMASA is much more time consuming, since the sole evaluation function requires as many operations as one iteration of DBA. However, there are many ways to speed up COMASA, if the time of execution is important. Firstly, the evaluation of a solution provides the coupling required for the computation of DBA, and/or conversely. This step could be factorized and would reduce the execution time by several factors. Secondly, many lower bounds of DTW have been developed in order to give a fast first trend of the result (see [30] for more details on lower bounds for DTW). Lower bounding DTW is shown to speed up significantly DTW-based algorithms.

[8] This improvement is computed using the geometric average since it is a mean of ratios between COMASA and NLAAF or DBA.



(a) A class of the *Trace* dataset.

(b) The average sequence of the class from the *Trace* dataset.

(c) A class of the *50words* dataset.

(d) The average sequence of the class from the *50words* dataset.

Figure 12: An example of the result of COMASA on one class of both “Trace” and “50words” datasets.

Table 2: This table presents the experiments carried out on the standard datasets. On the left: comparison of intraclass inertia under DTW between the medoid, NLAAF, DBA and COMASA. On the right: comparison of runtimes (in seconds) between DBA over ten iterations, and COMASA.

Dataset	Intraclass inertia				Runtime (s)	
	Medoid	NLAAF	DBA	COMASA	DBA	COMASA
50words	13.52	11.98	6.21	**4.71**	2	155
Adiac	0.22	0.21	0.17	**0.15**	1	71
Beef	31.24	29.90	9.50	**6.12**	4	292
CBF	14.25	15.35	13.34	**12.51**	7	981
Coffee	0.82	0.72	0.55	**0.47**	3	276
ECG200	9.52	11.34	6.95	**6.29**	1	150
FaceAll	19.04	17.77	14.73	**13.83**	4	520
FaceFour	38.10	34.46	24.87	**21.91**	5	485
Fish	1.39	1.35	1.02	**0.94**	15	1 069
GunPoint	9.55	7.24	2.46	**2.03**	3	283
Lighting2	109.35	194.07	77.57	**67.46**	36	318
Lighting7	41.28	48.25	28.77	**26.27**	3	278
OliveOil	0.023	0.018	0.018	**0.017**	7	500
OSULeaf	52.40	53.03	22.69	**19.55**	19	1 607
SwedishLeaf	2.52	2.50	2.21	**1.81**	2	171
Synthetic control	11.15	9.71	9.28	**8.70**	1	38
Trace	1.79	1.65	0.92	**0.60**	6	192
Two patterns	9.50	9.19	8.66	**7.42**	30	298
Wafer	81.05	54.66	30.40	**24.27**	118	1 367
Yoga	38.15	40.07	37.27	**11.10**	416	3 330

The best scores are shown in boldface.

6. Conclusion

Summarizing a set of sequences was mostly driven by the development of computational biology. This field was particularly interested in the multiple alignment problem.

There are actually two of use of a consensus sequence: computational and visualization. For both purposes, the comparison of sequences under time warping fits a commonly accepted definition of the time dimension. The choice of the distance is central for the analysis of time series. In most cases, disposing of a corresponding averaging method is essential.

We introduced the notion of compact multiple alignments and its use for the averaging of time series. This article presented the generalization of the theory introduced in [17] through the definition of compact multiple alignments. We have shown that COMASA achieves better results on all tested datasets, both visually and statistically.

We believe this work opens up a number of research directions. First, the study of the optimal length of the average sequence could raise compression perspectives. Actually, a short average sequence would be built around major states of evolution, providing a sampling of the data. Second, the adaptation of COMASA for symbolic sequences could give a new approach for multiple sequence alignment, with well-known applications in computational biology. Moreover, the field of computational biology could benefit from the connection between the consensus sequence and multiple alignments. We believe the g and f functions can be extended for symbolic sequences, providing supplement theory for multiple DNA/RNA sequence alignments. Finally, this research raises several questions on the topology of temporal spaces. When DTW is used to compare sequences, the embedding space of the sequences is then a semi-pseudometric space, which prevents the use of classical properties on Euclidean spaces. However, the average sequence induces L Euclidean spaces, around its constituting element. This rephrasing echoes the theory of manifolds. Studying the use of the average sequence to form a manifold from the time series could have important implications in dimensionality reduction.

Acknowledgments

The authors would like to thank the French Space Agency (CNES) and Thales Alenia Space for supporting this work under research contract n°1520011594. The authors would also like to thank the UCR group [18] for making available to the data mining community the time series datasets used in this article.

References

[1] D. Gusfield, Algorithms on Strings, Trees, and Sequences: Computer Science and Computational Biology, Cambridge University Press, 1997.

[2] H. Sakoe, S. Chiba, A dynamic programming approach to continuous speech recognition, in: Proceedings of the Seventh International Congress on Acoustics, Vol. 3, 1971, pp. 65–69.

[3] A. P. Shanker, A. Rajagopalan, Off-line signature verification using DTW, Pattern Recognition Letters 28 (12) (2007) 1407 – 1414.

[4] D. Sankoff, J. Kruskal, Time Warps, String Edits and Macromolecules: The Theory and Practice of Sequence Comparison, Addison Wesley Publishing Company, 1983, Ch. The symmetric time-warping problem: from continuous to discrete, pp. 125–161.

[5] J. Aach, G. M. Church, Aligning gene expression time series with time warping algorithms, Bioinformatics 17 (6) (2001) 495–508.

[6] Z. Bar-Joseph, G. Gerber, D. K. Gifford, T. S. Jaakkola, I. Simon, A new approach to analyzing gene expression time series data, in: RECOMB: Proceedings of the sixth annual international conference on Computational Biology, ACM, New York, NY, USA, 2002, pp. 39–48.

[7] D. M. Gavrila, L. S. Davis, Towards 3-D model-based tracking and recognition of human movement: a multi-view approach, in: IEEE International Workshop on Automatic Face- and Gesture-Recognition., 1995, pp. 272–277.

[8] T. Rath, R. Manmatha, Word image matching using dynamic time warping, in: IEEE Conference on Computer Vision and Pattern Recognition, Vol. 2, 2003, pp. 521–527.

[9] V. Niennattrakul, C. A. Ratanamahatana, Shape Averaging under Time Warping, in: International Conference on Electrical Engineering/Electronics, Computer, Telecommunications, and Information Technology, 2009.

[10] E. N. Gilbert, H. O. Pollak, Steiner minimal trees, SIAM Journal on Applied Mathematics 16 (1) (1968) 1–29.

[11] E. Dimitriadou, A. Weingessel, K. Hornik, A combination scheme for fuzzy clustering, International Journal of Pattern Recognition and Artificial Intelligence 16 (7) (2002) 901–912.

[12] V. Niennattrakul, C. A. Ratanamahatana, Inaccuracies of shape averaging method using dynamic time warping for time series data, in: S. Berlin (Ed.), Computational Science – ICCS, Vol. 4487 of LNCS, 2007.

[13] T. Liao, B. Bolt, J. Forester, E. Hailman, C. Hansen, R. Kaste, J. O'May, Understanding and projecting the battle state, in: 23rd Army Science Conference, 2002.

[14] T. W. Liao, C.-F. Ting, P.-C. Chang, An adaptive genetic clustering method for exploratory mining of feature vector and time series data, International Journal of Production Research 44 (2006) 2731–2748.

[15] L. Gupta, D. Molfese, R. Tammana, P. Simos, Nonlinear alignment and averaging for estimating the evoked potential, IEEE Transactions on Biomedical Engineering 43 (4) (1996) 348–356.

[16] S. Ongwattanakul, D. Srisai, Contrast enhanced dynamic time warping distance for time series shape averaging classification, in: International Conference on Interaction Sciences, ACM, 2009.

[17] F. Petitjean, A. Ketterlin, P. Gançarski, A global averaging method for dynamic time warping, with applications to clustering, Pattern Recognition 44 (3) (2011) 678–693.

[18] E. Keogh, X. Xi, L. Wei, C. A. Ratanamahatana, The UCR Time Series Classification / Clustering Homepage, http://www.cs.ucr.edu/~eamonn/time_series_data/ (2006).

[19] H. Sakoe, S. Chiba, Dynamic programming algorithm optimization for spoken word recognition, IEEE Transactions on Acoustics, Speech and Signal Processing 26 (1) (1978) 43–49.

[20] S. B. Needleman, C. D. Wunsch, A general method applicable to the search for similarities in the amino acid sequence of two proteins, Journal of Molecular Biology 48 (3) (1970) 443–453.

[21] L. Wang, T. Jiang, On the complexity of multiple sequence alignment, Journal of Computational Biology 1 (4) (1994) 337–348.

[22] C. Notredame, D. G. Higgins, SAGA: Sequence Alignment by Genetic Algorithm, Nucleic Acids Research 24 (8) (1996) 1515–1524.

[23] R. C. Edgar, MUSCLE: a multiple sequence alignment method with reduced time and space complexity, BMC Bioinformatics 5 (1) (2004) 1792–1797.

[24] J. Pei, R. Sadreyev, N. V. Grishin, PCMA: fast and accurate multiple sequence alignment based on profile consistency, Bioinformatics 19 (3) (2003) 427–428.

[25] T. Lassmann, E. L. L. Sonnhammer, Kalign - an accurate and fast multiple sequence alignment algorithm, BMC Bioinformatics 6 (1) (2005) 298–306.

[26] C. Notredame, D. G. Higgins, J. Heringa, T-coffee: a novel method for fast and accurate multiple sequence alignment, Journal of Molecular Biology 302 (1) (2000) 205–217.

[27] J. Pei, N. V. Grishin, PROMALS: towards accurate multiple sequence alignments of distantly related proteins, Bioinformatics 23 (7) (2007) 802–808.

[28] A. Kapsalis, V. J. Rayward-Smith, G. D. Smith, Solving the graphical steiner tree problem using genetic algorithms, The Journal of the Operational Research Society 44 (4) (1993) 397–406.

[29] A. W.-C. Fu, E. J. Keogh, L. Y. H. Lau, C. A. Ratanamahatana, R. C.-W. Wong, Scaling and time warping in time series querying., VLDB Journal 17 (4) (2008) 899–921.

[30] E. Keogh, C. A. Ratanamahatana, Exact indexing of dynamic time warping, Knowledge and Information Systems 7 (3) (2005) 358–386.

[31] J. H. Holland, Adaptation in Natural and Artificial Systems, MIT Press, Cambridge, MA, USA, 1975.

[32] D. E. Goldberg, Genetic Algorithms in Search, Optimization and Machine Learning, Addison-Wesley Longman Publishing Co., Inc., Boston, MA, USA, 1989.

[33] C. Darwin, On the origin of species by means of natural selection, London, John Murray, 1859.

[34] J.-B. Lamarck, Philosophie zoologique, Dentu, Paris, 1809.

[35] J. J. Grefenstette, Lamarckian learning in multi-agent environments, in: Proceedings of the Fourth International Conference on Genetic Algorithms, Morgan Kaufmann, 1991, pp. 303–310.

[36] J. Paredis, Coevolutionary life-time learning, in: Proceedings of the 4th International Conference on Parallel Problem Solving from Nature, PPSN IV, Springer-Verlag, London, UK, 1996, pp. 72–80.

[37] B. J. Ross, A Lamarckian Evolution Strategy for Genetic Algorithms, in: L. Chambers (Ed.), Practical Handbook of Genetic Algorithms: Complex Coding Systems, Vol. 3, CRC Press, 1999, pp. 1–16.

[38] D. E. Goldberg, S. Voessner, Optimizing global-local search hybrids, in: Genetic and Evolutionary Computation Conference, 1999, pp. 212–219.

[39] J. Baldwin, A new factor in evolution, American Naturalist 30 (1896) 441–451.

[40] L. D. Whitley, V. S. Gordon, K. E. Mathias, Lamarckian evolution, the baldwin effect and function optimization, in: Parallel Problem Solving from Nature, PPSN III, Springer-Verlag, London, UK, 1994, pp. 6–15.

[41] K. Ku, M. Mak, Exploring the effects of lamarckian and baldwinian learning in evolving recurrent neural networks, in: IEEE International Conference on Evolutionary Computation, 1997, pp. 617–621.

[42] P. Turney, Myths and legends of the baldwin effect, in: Workshop on Evolutionary Computation and Machine Learning at the 13th International Conference on Machine Learning, 1996, pp. 135–142.

Appendix A. Evolution theories

This section presents the main evolution theories.

Genetic algorithms (GAs) were proposed as a way to solve problems when there are no other computational tractable algorithms [31]. GAs are heuristic searches and optimization techniques inspired by natural evolution [32]. A GA operates on a population of artificial genes where each gene represents a potential solution to the problem. A solution can be evaluated through a fitness which is a measure of how this solution is "good". The algorithm carries out a process of evolution which consists of selecting and recombining genes to produce successive populations.

A variety of mechanisms behind the natural evolution process have been suggested and each one of them constitutes a theory of evolution. Let us briefly review three of them to highlight their founding hypothesis and their impact on terms of evolutionary algorithms.

Appendix A.1. The Darwinian theory [33]

The Darwinian theory relies on three characteristics: (1) individuals are different from each others; (2) individual characteristics are inherited; (3) an individual adapted to its environment has more offspring than an individual that is not.

This theory is transposed in GAs in the following way: at each life-cycle (each generation), the solutions are computed from genotypes. All these solutions are evaluated. Then, the best individuals (according to the evaluation) are selected and their genetic material is recombined to produce the next generation.

Appendix A.2. The Lamarckian theory [34]

The Lamarckian theory relies on four characteristics: (1) individuals are different from each others; (2) individuals evolve to be more adapted to the environment; (3) acquired characteristics are inherited; (4) an individual adapted to its environment has more offspring than an individual that is not. Although in the natural life this thesis has been outclassed by the Darwinian theory, it has been successfully applied in artificial learning [35–38].

This theory is transposed in GAs in the following way: at each generation, the solutions are computed from the genotype. The computation of the solutions can modify the genotype through a local search method performed on the evaluation, whereas the Darwinian approach. All these solutions are evaluated. The best individuals (according to the evaluation) are selected and their *new* genetic material is recombined to produce the next generation.

Appendix A.3. The Baldwin effect [39]

The theory of Baldwin is similar to the Darwinian theory (no inheritance of acquired characteristics) but it introduces the phenotypic plasticity into this model. Phenotypic plasticity is defined as the organism flexibility and creativity to adapt its behavior to the environment throughout its lifetime: the higher the phenotypic plasticity, the more chances the individual will have to adapt itself to the environment throughout its lifetime, and thus to provide "a better solution".

This idea can transposed in GAs design by the more a genotype predisposes an individual to adapt itself, the more its genes will be transmitted and shared. At each generation, the solutions are computed from genotypes. Like in the Lamarckian approach, these computations can modify the genotypes. All these solutions are evaluated. Then, the best individuals (according to the fitness) are selected and their *initial* genetic material is used to produce the individuals of the next generation [40–42].