Detecting Time Series Motifs Under Uniform Scaling

Dragomir Yankov, Eamonn Keogh, Jose Medina, Bill Chiu, Victor Zordan
Dept. of Computer Science & Eng.
University of California, Riverside, USA
{dyankov, eamonn, medinaj, bill, vzb}@cs.ucr.edu

ABSTRACT

Time series motifs are approximately repeated patterns found within the data. Such motifs have utility for many data mining algorithms, including rule-discovery, novelty-detection, summarization and clustering. Since the formalization of the problem and the introduction of efficient linear time algorithms, motif discovery has been successfully applied to many domains, including medicine, motion capture, robotics and meteorology.

In this work we show that most previous applications of time series motifs have been severely limited by the definition's brittleness to even slight changes of uniform scaling, the speed at which the patterns develop. We introduce a new algorithm that allows discovery of time series motifs with invariance to uniform scaling, and show that it produces objectively superior results in several important domains. Apart from being more general than all other motif discovery algorithms, a further contribution of our work is that it is simpler than previous approaches, in particular we have drastically reduced the number of parameters that need to be specified.

Categories and Subject Descriptors

H.2.8 [**DATABASE MANAGEMENT**]: Database Applications —*Data mining*

General Terms

Algorithms

Keywords

Time Series, Motifs, Random Projection, Uniform Scaling

1. INTRODUCTION

Time series motifs are approximately repeated patterns found within the data. For many data mining areas the detection of such repeated patterns is of essential importance. A few tasks, among others, that utilize time series motif detection are for example rule-discovery [21], novelty-detection, clustering and summarization [26]. Motif discovery has been successfully applied throughout a large range of domains too, such as medicine [2][3], motion-capture [8][22], robotics, video surveillance [13] and meteorology [21]. Here, we show that the existing approaches for motif detection are limited to discovering pattern occurrences of the same length, failing to capture similarities when the occurrences are uniformly scaled along the time axis. To motivate the need for such uniform-scaling invariant motif discovery we will examine a synthetic time series (synthetic data is used here for ease of exposition, real-world examples are studied in the experimental section). Consider the time series in Figure 1.

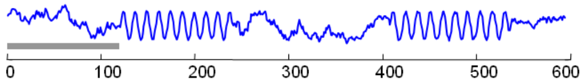


Figure 1: A time series of length 600. What is the best motif of length 120 (the length of the gray bar)?

If we are asked to point out the best repeated pattern of length 120, the answer appears trivial: there is an obvious repeated sine wave of approximate length 120 in two locations. However, as we can see in Figure 2, this is not the true motif in this data set.

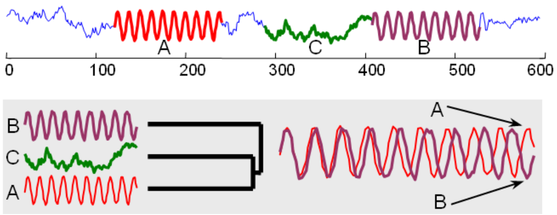


Figure 2: *Top*: An annotated version of Figure 1. *Bottom Left*: The closest subsequence to **A**, using Euclidean distance, is in fact subsequence **C**. *Bottom Right*: By plotting **B** on top of **A** we can see the reason for this unintuitive result.

The reason for this unintuitive result is that **A** differs from **B** by a linear scaling of 5%. This means that although shorter subsections of the two subsequences are almost identical, when we attempt to align them, the cumulative error of the out-of-phase sections will tend to dominate. Note

KDD'07, August 12–15, 2007, San Jose, California, USA.
Copyright 2007 ACM 978-1-59593-609-7/07/0008 ...$5.00.

that the problem here is exacerbated by the fact that the patterns are complex, meaning that they have many peaks and valleys, insuring that if two otherwise similar patterns have different linear scaling, at least some peaks will have to map to valleys, resulting in a large Euclidean distance. Motif discovery has been shown to be useful in many domains, but practitioners have only used it for relatively simple patterns (one or two peaks and valleys).

It is important to dismiss two apparent solutions to this problem before introducing our technique:

- *Why not replace the Euclidean distance with the Dynamic Time Warping (DTW) distance?* While DTW is a very useful tool for many data mining problems, it is not the solution here. For example, if we have a subsequence of length 500 that contains 10 heartbeats, and another subsequence of length 500 that contains 9 heartbeats, DTW is no more useful than Euclidean distance, because DTW must match *every* data point in each sequence, and there is no meaningful way to map 9 heartbeats to 10 heartbeats. What is required is *uniform scaling*, which compares the original 500 data points to a range of possible data points, say from 500 to 600, incorporating the second sequence.

- *Why not search for shorter patterns, and after finding the shorter motifs, somehow "grow" them with invariance to uniform scaling?* This idea does seem attractive initially. In the example in Figure 2, if we shorten the required pattern length to 100 instead of 120, we do find a subsection of A and a subsection of B to be the best motif. The problem is that in most domains, if we reduce the length of patterns of interest, the number of motifs will increase exponentially, and post processing all these false alarms will require considerable effort. To see this, consider the analogue of discrete motifs in text. This paper has a motif of the words "*Hominidae*" and "*Homininae*" with a hamming distance of just one. We could try to discover this motif by enumerating all motifs of length 4 with a hamming distance less than two, and seeing which ones we could grow to length 9. However there are thousands of false alarms, including "*well*"/"*will*", "*tool*"/"*toll*", "*moti(f)*"/"*moti(on)*" etc. Thus the idea of "growing" motifs (under invariance to "noise") appears unworkable in both real and discrete domains.

In this work we will show that we can efficiently discover motifs under uniform scaling. Furthermore, we will show on diverse datasets that accounting for uniform scaling allows us to discover motifs which are objectively and subjectively correct, and which are missed by the other time series motif discovery algorithms.

2. RELATED WORK

The importance of uniform scaling for indexing and matching time series has been noted and addressed in several communities, including motion-capture [16] and music. However, it has yet to be addressed for the motif problem. The idea that approximately repeated patterns can be informative has permeated the field of bioinformatics for decades [20], and was hinted at in time series data mining literature for as perhaps as long. However, the first formal definition of time series approximately repeated patterns, the time series motif, appeared as recently as 2003 [9]. Since then, there have been dozens of papers that use time series motifs in a variety of applications (hereafter we will use *time series motifs* and *motifs* interchangeably).

Garbay and colleagues have used time series motifs in different medical applications, including medical telepresence [10] and intensive care monitoring. Minnen et al. use motifs in a series of papers that examine the utility of monitoring on-body sensors [22]. Tanaka et al. use motifs to find patterns in sports motion capture data [28]. Murakami et al. use motifs to discover regularities in the behavior of robots [23]. In a series of papers Abe and colleagues [1] have used motifs as an input to rule finding algorithms in medical domains. Androulakis and colleagues use motifs for a variety of bioinformatics problems, for example finding "regimes with similar kinetic characteristics" in autoignition [2], and selecting maximally informative genes [3]. Hamid et al. use motifs as a primitive in a video surveillance application [13]. Celly and Zordan [8] use motifs to create Animated People Textures (APT) from motion-capture recordings; Applications for APT include video based animations for electronic games and creating background elements and special effects for movies.

Arita et al. [5] use time series motifs in a "real-time human proxy" (i.e telepresence) application. The idea is that motif discovery is used offline to capture typical human motions, for example "pointing with right hand" or "head nodding". These complex actions can then be represented and efficiently transmitted as a single symbol, rather than a complex set of real valued motion trajectories. The authors do understand the need for uniform scaling in their application. This is achieved by a combination of human intervention and a quadratic time dynamic programming technique.

In spite of this wealth of works that use or extend the notion of time series motifs, none of them address the uniform scaling problem. Perhaps the most sophisticated extension to the original motif paper is [28], yet even this work explicitly states "We can not extract motifs whose lengths are different from each other though they have the same behavior". This is exactly the problem solved in this work.

3. SIMILARITY UNDER UNIFORM SCALING

We start by briefly revising the uniform scaling distance as introduced in [18]. Suppose that we have a query time series $\mathbf{Q} = (q_1, q_2, \ldots, q_{m_q})$ and a candidate matching sequence $\mathbf{C} = (c_1, c_2, \ldots, c_{m_c})$. Without loss of generality, assume that $m_q \leq m_c$. A uniformly scaled version of the query $\mathbf{Q}$ with scaling factor $\frac{m_q}{s}$ is the time series $\mathbf{Q}^s = (q_1^s, q_2^s, \ldots, q_s^s)$, where $q_i^s = q_{\lceil i \frac{m_q}{s} \rceil}$ (see Figure 3).

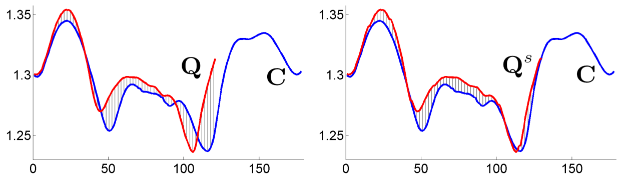


Figure 3: Comparing $\mathbf{Q}$ and $\mathbf{C}$ directly (*Left*) yields larger Euclidean distance. *Right*: Stretching $\mathbf{Q}$ with 8% produces $\mathbf{Q}^s$ which resembles very closely the prefix of $\mathbf{C}$.

The uniform scaling distance $d_u(\mathbf{Q}, \mathbf{C})$ is defined as the optimal squared Euclidean distance between some prefix of $\mathbf{C}$ with length $s \geq m_q$ and the query scaled to the size of that prefix. Or more formally:

$$d_u(\mathbf{Q}, \mathbf{C}) = \min_{m_q \leq s \leq m_c} \sum_{i=1}^{s} (q_i^s - c_i)^2 \qquad (1)$$

To check all s, equation (1) computes $(m_c - m_q + 1)$ Euclidean distances between the scaled query and a prefix of $\mathbf{C}$. As for most practical purposes a very small rescaling is required to identify the best match (see Section 6), d_u has an overall amortized cost of $\Theta(cm_q)$, for some constant $c > 1$ [18]. With this in mind, one can think of the nearest neighbor search under uniform scaling as a search under Euclidean distance but in a new, denser sample space. In this space every original candidate sequence $\mathbf{C}$ is replaced with a neighborhood of c new sequences.

4. NEAREST NEIGHBOR MOTIFS

Now, we formally introduce the time series motif finding problem. As searching for motifs in the denser uniform scaling space can be hard under the original motif definition, we also provide an equivalent alternative definition that will be utilized in the proposed approach.

Time series motifs are defined as the *approximate* occurrences of a subsequence in the time series $\mathbf{S} = (s_1, s_2, \ldots, s_n)$ at several *significantly different* positions [19]. "Approximate" here is expressed in terms of a distance function d and a range r, i.e. two subsequences $\mathbf{S}_i^m = (s_i, s_{i+1}, \ldots, s_{i+m-1})$ and $\mathbf{S}_j^m = (s_j, s_{j+1}, \ldots, s_{j+m-1})$ are approximately similar if $d(\mathbf{S}_i^m, \mathbf{S}_j^m) \leq r$. The starting positions i and j are assumed to be significantly different, if there exists i_1 such that $i < i_1 < j$ and $d(\mathbf{S}_i^m, \mathbf{S}_{i_1}^m) > r$. This ensures that while looking for approximate appearances of a subsequence we are not considering its slightly shifted versions, which often will be within range r from it. Two approximately similar, with respect to r and d, subsequences that start at significantly different positions are called a *non-trivial match.*

Using the introduced terminology, the formal definition of a motif is as follows [19]:

Definition 1. Range Motif. Given a time series S, a subsequence length m and a range r, the most significant range motif of length m in S is the subsequence $\mathbf{S}_i^m$, which has the highest count of non-trivial matches $\mathbf{S}_j^m$, such that $d(\mathbf{S}_i^m, \mathbf{S}_j^m) \leq r$.

One could also be interested in the subsequence that has the second largest count of similar subsequences. This subsequence is called second-motif. Other, less significant motifs, can also be defined similarly.

There exists a close analogy between motif detection based on Definition 1 and density estimation methods using neighborhood of fixed volume [11]. Indeed, detecting a time series motif is very similar to computing the density around each subsequence, where the examples in the subsequence neighborhood are its non-trivial matches. The most significant motif is simply the sequence whose neighborhood has maximum density (i.e. largest number of neighbors).

One of the problems of the fixed range neighborhood, however, is that specifying the right range is not always intuitive [11]. Consider for example Figure 4. Even a person of expertise may not be able to provide the right range to detect the motif in the time series, as different time series may be sampled at different rate and hence require different ranges r. Therefore, probing a number of possible ranges is inevitable. In this process, very small ranges will result in empty neighborhoods, and no motif detection, while large ranges will return as most significant some not so interesting patterns. As pointed out in Section 3, allowing a uniform scaling factor will further increase the number of examples and the probability of having higher density regions, i.e. small neighborhoods populated with many examples. Thus, the task of specifying appropriate ranges for detecting the most significant motif in the uniformly scaled space becomes rather difficult.

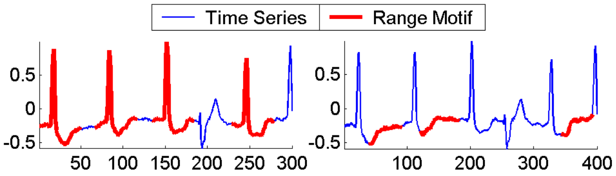


Figure 4: The same time series sampled at different rate. Using the same range r, one detects two different motifs.

Here, we express the concept of significant motifs in terms of nearest neighbors. The k-Nearest-Neighbor (kNN) method provides an alternative approach to nonparametric density estimation. It is more adaptive to variations in the space density and alleviates the problem of specifying the not so intuitive range parameter. Firstly, we introduce the term non-trivial nearest neighbor as:

Definition 2. Non-trivial nearest neighbor. Let $\mathbf{S}_i^m$ and $\mathbf{S}_j^m$ be two subsequences of length m in a time series $\mathbf{S}$ of length n. We say that $\mathbf{S}_j^m$, $1 \leq j \leq n$–m+1 is a non-trivial neighbor of $\mathbf{S}_i^m$, $1 \leq i \leq n$–m+1, if there exists a subsequence $\mathbf{S}_{i_1}^m$ in $\mathbf{S}$, such that $i < i_1 < j$ and $d(\mathbf{S}_i^m, \mathbf{S}_{i_1}^m) > d(\mathbf{S}_i^m, \mathbf{S}_j^m)$. The nearest non-trivial neighbor to $\mathbf{S}_i^m$, is the non-trivial neighbor $\mathbf{S}_j^m$ with minimal distance $d(\mathbf{S}_i^m, \mathbf{S}_j^m)$.

Similarly, one can define the second or another more remote non-trivial nearest neighbor. Using the k non-trivial nearest neighbors to a sequence we come with a formalization which, equivalently to Definition 1, captures the notion of approximately similar patterns in the time series data.

Definition 3. Nearest-Neighbor Motif. The most significant nearest neighbor motif of length m in a time series $\mathbf{S}$ of length n, is the subsequence $\mathbf{S}_i^m$, $1 \leq i \leq n$–m+1 which has minimal distance to its non-trivial nearest neighbor $\mathbf{S}_j^m$, $1 \leq j \leq n$–m+1.

The k^{th} most significant motif is now defined as the sequence with minimal average distance to its k nearest neighbors. Definitions 2 and 3 can naturally be extended for motifs under uniform scaling if the length of the matching sequence $\mathbf{S}_j^m$ is allowed to differ from the length of $\mathbf{S}_i^m$, and if the Euclidean distance $d(\mathbf{S}_i^m, \mathbf{S}_j^m)$ is replaced with the uniform scaling distance d_u:

Definition 4. Uniform Scaling Motif. The most significant uniform scaling motif of length m_q, in a time series $\mathbf{S}$ of length n, is the subsequence $\mathbf{S}_i^{m_q}$, $1 \leq i \leq n$–m_q+1 which has minimal uniform scaling distance $d_u(\mathbf{S}_i^{m_q}, \mathbf{S}_j^{m_c})$ to its non-trivial nearest neighbor $\mathbf{S}_j^{m_c}$, $1 \leq j \leq n–m_c+1$.

For the rest of the paper we derive an effective and efficient probabilistic approach to detecting the best motifs under uniform scaling, and also show that such motifs often represent far more “interesting” patterns in the data, than the motifs under Euclidean distance.

5. PROBABILISTIC MOTIF DETECTION

For a time series $\mathbf{S}$ of length n the brute force motif detection algorithm will perform $\Theta(n^2)$ pairwise distance computations, between subsequences $\mathbf{S}_i^{m_q}$ and $\mathbf{S}_j^{m_c}$. As discussed in Section 3, if each of these computations is performed with uniform scaling, then they will have a complexity of $\Theta(cm_q)$. This means that the total cost of finding the most significant motif under uniform scaling, using a brute force search, will become $\Theta(cn^2 m_q)$. Lower bounding techniques, such as the one suggested in [18], can speed up the computation of the uniform scaling distances. For larger values of n, however, the algorithm still remains intractable.

Rather than computing all pairwise distances, the approach proposed here runs a filter linear in n, and removes from consideration a large number of subsequences. In a following refinement step, only the uniform scaling distance between the nonfiltered sequences is computed. The filtering step is derived from the random projection algorithm proposed by Buhler and Tompa [6]. Though probabilistic, we demonstrate that the approach has a high motif detection rate. The gain in speed-up over the brute force search, however, is enormous.

5.1 The Random Projection Algorithm

The following “challenge” problem, introduced by Pevzner and Sze [24], has inspired much endeavor in the bioinformatics community since it appeared few years ago: Given is a sample of $t = 20$ nucleotide strings of length $n = 600$. An unknown motif M of length $l = 15$ is used to generate t new motif occurrences. Every occurrence differs from M in exactly $d = 4$ base pairs (letters). Each one of these motif occurrences is *planted* at a random location in one of the t nucleotide strings. The goal is to detect the planted (l, d)-motif.

Detecting the hidden signal turns out to be rather difficult as two occurrences might have as many as eight differences, which disguises the original motif considerably. While deterministic solutions of the problem are exponential in the motif length l, and hence impractical, Pevzner and Szi show that approaches, such as sampling or expectation maximization detect local minima while searching for the (15,4)-motif.

In [6], Buhler and Tompa demonstrate an interesting probabilistic approach that efficiently identifies the challenge motif as well as other difficult motifs, e.g. the (14,4)-, (16,5)-, and (18,6)-motifs. They study the family of *locality-preserving* hash functions $h(\mathbf{w})$ [14], that project the l-letter words $\mathbf{w}(w_1, w_2, \ldots, w_l)$ over an alphabet Σ, into $\hat{l}$-letter words $(\hat{l} < l)$. I.e. $h(\mathbf{w}) : \Sigma^l \to \Sigma^{\hat{l}}$, and $h(\mathbf{w}) = \widehat{\mathbf{w}}(w_{i_1}, w_{i_2}, \ldots, w_{i_{\hat{l}}})$. The basic observation is that if we select $\hat{l} \leq l-d$ positions at random from all l-letter subsequences in the strings, there is high probability that at least $\epsilon > 1$ occurrences of the planted motif will hash to the same bucket (string). On the contrary, all other l-letter strings are likely to hash into buckets with less than ϵ elements. The algorithm derived by Buhler and Tompa, called PROJECTION, performs I iterations, repeatedly selecting a different hash function h_i using a random set of projecting dimensions $\{w_i\}$. After hashing all l-letter subsequences, the threshold ϵ is applied to filter out buckets that are unlikely to contain the motif occurrences. Finally, a refinement step based on expectation maximization infers the motif that would maximize the likelihood of observing the unfiltered buckets from each iteration.

Using a random set of dimensions for hashing by PROJECTION is reasonable, as it guarantees that a set of random words over the alphabet Σ will be uniformly spread across all possible hash values. Here, of course, an implicit assumption is made, that all letters in the alphabet are equiprobable. Another limitation of locality-preserving hashing is that it is effective for a relatively small number of projected dimensions (10∼20) [14]. Applying it to larger subsequences would practically require all pairwise subsequence comparisons to be performed.

5.2 PROJECTION For Time Series Motifs

PROJECTION can be adapted to detect time series range motifs (see Definition 1), provided there is a suitable representation of the real value data with a finite alphabet. Chiu et al. [9] demonstrate one such adaptation. They first convert all extracted time series subsequences to a Symbolic Aggregate approXimation (SAX) form [19] (Figure 5).

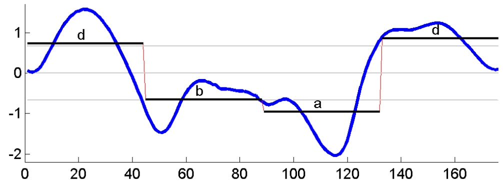
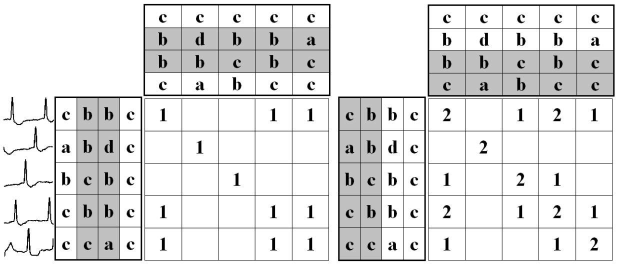


Figure 5: Symbolic aggregate approximation of a time series as the four letter word “dbad”. The size of the alphabet used is 4 ($\Sigma = \{$a, b, c, d$\}$). The grey horizontal cut-lines outline equal volume segments under the normal distribution curve.

SAX normalizes every sequence S^m to have mean zero and standard deviation one. Then, using a user specified word length l, it computes its piecewise aggregate approximation $PAA(S^m) = \bar{S}(\bar{s}_1, \bar{s}_2, \ldots, \bar{s}_l)$, where $\bar{s}_i = \frac{l}{m} \sum_{j=\frac{m}{l}(i-1)+1}^{\frac{m}{l}i} s_j$. Finally it splits the area under the normal curve into $|\Sigma|$ segments, where $|\Sigma|$ is a user specified alphabet size. Every value $\bar{s}_i$ in the PAA representation is then substituted with the letter w_i, which labels the corresponding normal curve segment. This operation transforms the initial sequence S^m into the l-letter word $\mathbf{w}$.

Assigning letters to segments of equal volume under the normal curve guarantees the implicit assumption of PROJECTION for equiprobable symbols. Therefore, all collisions during the hashing process are most likely a result of the similarities between certain sequences.

The time series motif discovery continues by building a sparse collision matrix M_{nxn} (see Figure 6). At each iteration of the projection algorithm, when two sequences converted to the SAX words $\mathbf{w}_x$ and $\mathbf{w}_y$ are subsequently projected to the same value $h_i(\mathbf{w}_x) = h_i(\mathbf{w}_y) = \widehat{\mathbf{w}}_x$, a counter



	c b b c	c d b a	c b c b	c b b c	c a c c
c b b c	1			1	1
a b d c		1			
b c b c			1		
c b b c	1			1	1
c c a c	1			1	1

	c b b c	c d b a	c b c b	c b b c	c a c c
c b b c	2		1	2	1
a b d c		2			
b c b c	1		2	1	
c b b c	2		1	2	1
c c a c	1			1	2

Figure 6: *Left*: Iteration 1 of the projection algorithm. Dimensions 1 and 4 are selected as projecting dimensions. The corresponding positions for the identically projected strings in the collision matrix are increased with one. *Right*: The collision matrix after the second iteration.

for cell (x, y) in M is incremented by one. At the end of the algorithm, all cells in M with counters bigger than the threshold ϵ are returned as possible locations of the most significant motifs. For these, and only these, locations the algorithm computes the actual Euclidean distance between their corresponding sequences. The algorithm, though subquadratic in theory, is empirically demonstrated in [9] to be approximately linear with respect to the time series size n both in terms of memory requirements and in terms of the number of brute force Euclidean distance computations.

6. LEARNING MOTIFS UNDER UNIFORM SCALING

A significant drawback of the above time series motif finding approach is that it requires supervision in selecting a large set of input parameters. Namely, the tuple ($|\Sigma|$, m, l, d, $\hat{l}$, I), with elements respectively the alphabet size ($|\Sigma|$), the minimum time series motif length (m), the length of its string representation (l), the number of letter differences allowed between two motif occurrences (d), the number of hashed dimensions ($\hat{l}$), and the number of iterations necessary to detect the motif (I). This differs from the original PROJECTION algorithm, which assumes that all strings are over the four-letter DNA alphabet, and that the motif has fixed size l and number of differences d (e.g. for the challenge problem l=15 and d=4). Therefore, what needs to be estimated is only the tuple ($\hat{l}$, I)[1].

Here we show that the time series motif projection algorithm of [9] can be extended to capture motifs under the uniform scaling distance d_u (see Definition 3). We further demonstrate that an optimal, in terms of effectiveness and efficiency, tuple ($|\Sigma|$, l, d, $\hat{l}$, I) can be learned off-line in a completely unsupervised manner, requiring the user to provide only the minimum time series motif length m.

We first make two observations that allow for the efficient adaptation of the uniform scaling distance in the random projection algorithm. The first observation suggests that one can eliminate large scaling factors in the computation of the distance, as they lead to excessive stretches of $\mathbf{S}_i^{m_q}$, that cannot result in a significant motif. The maximal scaling factor, evaluated on a number of real-world data sets, which impacts the accuracy is 10% $\sim$ 40%. The second observation is that there is no need to check every scaling factor either, as close scaling factors produce similar results. For example, if we have stretched $\mathbf{S}_i^{m_q}$ 5% and identified the motifs, next we might skip stretching it 6% as with high probability the most significant motifs across the data set remain the same.

To compute the maximum scaling factor that is sufficient to detect any possible motifs for a data set, we study the cumulative empirical distribution of the scalings on the training data. Suppose that on the training set the probability that the most significant motif requires $x\%$ scaling is p_x. The cumulative distribution $P(x \leq k) = \sum_{x=0}^{k} p_x$ now gives us the scaling factor k that is sufficient to detect the best motif under uniform scaling with a high confidence (see Figure 7 left). The results are computed on the *brain activity* data set discussed in the next section and are consistent with results from the other data sets too.

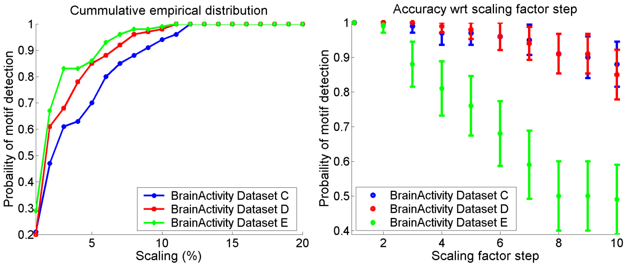


Figure 7: *Left:* Cumulative distribution of the maximal scaling factor that can impact the most significant motif. Large scaling factors (>20%) seldomly produce a significant motif. *Right:* Checking every single scaling may not be necessary. For some data sets checking every second (or third) factor still leads to detecting the most significant motif with very high probability.

In the experiments here we impose the constraint $P(x \leq k) = 1$, i.e. we check all scalings that have produced a significant uniform scaling motif for any of the training time series. As seen from Figure 7, the maximum scaling factor k that can impact the most significant motif is relatively small. The heuristic holds across different motif lengths m_q too. We further observe that increasing m_q leads to a decrease in k. This can intuitively be explained with the fact that larger motif lengths define a higher dimensional, and hence sparser, space where all nearest neighbors start drifting apart.

An evaluation of the second observation is presented in Figure 7 right (the statistics are again computed for the *brain activity* data). The graph shows that if, for example, the scaling factor is increased by 4% at a time, on some data sets (e.g. *Dataset C* and *Dataset D*), one can still detect more than 95% of the most significant motifs. The fact, again inferred from the training data, can be used in optimizing the search for other unseen time series generated from a similar process. Combining the two results allows us to achieve speed-up that makes the uniform scaling distance computation comparable to the computation of the Euclidean distance.

To cope with the requirement of supervision in the algorithm's parametrization, we look at some of the properties of the space defined by the projected sequences.

Let us assume that the tuple ($|\Sigma|$, l) is fixed. Using these

[1] Both algorithms also require the filtering threshold ϵ, but the authors show that low values as $\epsilon = 1$ or 2 are reasonable and perform satisfactorily.

alphabet and word sizes, we apply SAX to map the best motif occurrences $S_i^{m_q}$ and $S_j^{m_c}$ for a time series t into the equal length words $\mathbf{w}_i$ and $\mathbf{w}_j$ respectively. If d_t is the Hamming distance between these two words, then the maximum Hamming distance on the training set is: $d = \max_t d_t$. For a particular projection size $\hat{l} \leq l - d$, with analysis similar to the one in [6], we obtain the lower bounding probability of any two motifs in the training set to be hashed in the same bin:

$$p_d = \frac{\binom{l-d}{\hat{l}}}{\binom{l}{\hat{l}}} \tag{2}$$

We would like to perform I iterations of PROJECTION and guarantee that at least one of the functions h_i hashes together the motif occurrences for any time series t in the training set. In I independent trials the probability that none of the hashing functions detects the most significant motif, expressed as a function of the tuple $\theta = (|\Sigma|, l, d, \hat{l}, I)$, is:

$$p(\theta) = (1 - p_d)^I \tag{3}$$

There are several important observations to point out here. Firstly, increasing I minimizes $p(\theta)$, and hence the chance of omitting a motif. Secondly, the probability p_d is monotonically increasing when decreasing the projection size $\hat{l}$. Therefore, decreasing $\hat{l}$ also minimizes $p(\theta)$. However, the large number of iterations I and the small projection sizes $\hat{l}$ also increase the number of dissimilar sequences that would hash together. All those false positive pairs will not be filtered by the algorithm, and the actual distance between the sequences will be evaluated.

The two objectives that we try to optimize now are the effectiveness and the efficiency of the system. On one hand, to obtain an effective system, that produces very few false dismissal while searching for the most significant motifs, one needs to minimize $p(\theta)$. On the other hand, the efficiency of the system depends on how many iterations I are necessary to be performed and how many false positives are also returned for the subsequent refinement. As each iteration goes through all n subsequences, the cost of the hashing operations is $\Theta(In)$. If E_0 is the number of sequences that are less than or equal to d symbols apart, and E_i the number of sequences that are $d + i$ symbols apart, then the term $\sum_{i=0}^{l-1} E_i[1 - (1 - p_{d+i})^I]$, estimates the expected number of brute force comparisons to be performed during the refinement step. To summarize, we derive the following discrete optimization problem, where the effectiveness objective is expressed as a constraint:

$$\text{minimize} : \quad \mathcal{L}(\theta) = In + \sum_{i=0}^{l-1} E_i[1 - (1 - p_{d+i})^I] \tag{4}$$

$$\text{subj. to} : \quad p(\theta) \leq q \tag{5}$$

From constraints (5) the number of iterations I can be expressed as a function of ($|\Sigma|$, l, $\hat{l}$). Namely, $I \log(1 - p_d) \leq q$ which yields $I(|\Sigma|, l, \hat{l}) = \lceil \frac{\log q}{\log(1-p_d)} \rceil$. In the experiments we set $q = 0.05$, which together with the requirement that the two before mentioned observations should hold with probability higher than 0.95, guarantees that the probability of detecting the best motifs for all time series in the training set is at least 0.9.

After substituting I in (4), $\mathcal{L}(\theta)$ is minimized by performing a full search in the discrete space defined by $|\Sigma| \in [2, 10]$, $l \in [2, 20]$ and $\hat{l} \in [2, l - d]$, where the intervals are determined based on the bounds within which the locality preserving hashing and the SAX algorithm are known to be effective. Note, that the search is over a relatively small space of discrete tuples, which allows for the training procedure to identify the optimal parametrization in reasonable time (within minutes for our experiments), and requires no supervision at all.

7. EXPERIMENTAL EVALUATION

We demonstrate the usefulness of the uniform scaling distance for detecting motifs in three real-world data sets - brain activity data, motion capture time series, and time series extracted from the shapes of projectiles[2]. The data sets are selected from diverse domains, and are with different characteristics as periodicity, amount of noise, approximate length of the available motifs, and also variable length of the time series for the example.

An evaluation for the expected number of false dismissals, introduced by the probabilistic scheme, as well as the average speed-up over the brute force approaches is also presented.

7.1 Motifs in Brain Activity Time Series

Physiological data, such as respiratory recordings, heartbeats or brain waves, often contain scaled motifs that are indicative either of normal activities or of certain preconditions. Here we study three data sets of brain wave recordings from epileptic patients [4] (see Figure 8). The time series in *Dataset E* were recorded during epileptic attacks, while *Dataset C* and *Dataset D* contain recordings from seizure-free periods.

Finding the most significant motif in the data can be valuable for one of the primary tasks in the field, namely predicting the seizure periods. The assumption here is that very similar patterns are likely to have similar continuations, which can be used in forecasting the unobserved outcomes [27]. On the other hand, the strong similarity in motif occurrences in seizure data as E, can help in isolating groups of neurons that trigger identical activity during the epileptic attacks.

Each data set contains 100 time series of length 4096 points. We split them into a training and testing set of 50 time series each. Note that the time series were recorded from different patients, and still similar patterns can be observed across the patients, which is essential for the learning procedure to be effective. As the low frequency noise can accumulate quickly and disguise the real patterns, we further run a low pass moving average filter of size ten data points to smooth the data. Both, the Euclidean distance and the uniform scaling distance find some subjectively interesting motifs, yet the uniform scaling motifs are often more meaningful visually. For example, noisy “plateaus” are often detected by the Euclidean distance (see the graph for *Data set D*), whereas the uniform scaling usually identifies bursts or periodic patterns which are more useful for better diagnosis

[2] All data used in the evaluation is available at `http://www.cs.ucr.edu/~eamonn/SIGKDD07/UniformScaling.html`

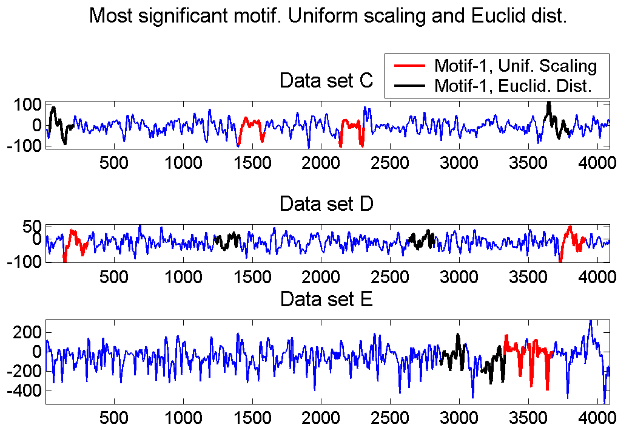


Figure 8: *Brain-wave data sets.* Motifs discovered by simple Euclidean distance and with uniform scaling. The sliding window length is set to $m_q = 174$ (recordings for 1 sec.). The detected motif occurrences for C, D and E have lengths m_c equal to 184, 185 and 192 respectively. The uniform scaling motifs are often more interpretable and visually meaningful.

and treatment.

For the three data sets the maximal scaling that can impact the most significant motif has been found to be around 15% (see Section 6). The best tuple ($|\Sigma|$, l, $\hat{l}$, I) for the random projection algorithm, learned also on the training data, is given in Table 1. For every corresponding tuple, the table also summarizes the average test accuracy of the method when every single scaling factor is checked (column 1); and when every second or third scaling factor is checked (columns 2 and 3). The results are consistent with the expectation computed on the training set (see Figure 7), and show that a speed-up can be obtained in computing the uniform scaling distance by checking every third (Data sets C and D) or every second (Data set E) scaling factor and still obtain relatively high accuracy rate in the detected motifs.

Table 1: Optimal parameters for the three data sets and motif detection test accuracy with respect to the scaling factor step.

	($\|\Sigma\|$, l, $\hat{l}$, I)	step=1	step=2	step=3
Data set C	(4, 7, 6, 45)	0.90	0.90	0.90
Data set D	(6, 10, 5, 35)	1.00	0.88	0.86
Data set E	(3, 10, 8, 31)	0.92	0.92	0.7

We also compute the average speed improvement for a single scaling factor. Figure 9 left presents the improvement introduced by random projection in terms of the number of performed operations, as compared with the brute force searching algorithm. For all three data sets, the probabilistic method performs less than 1% of the operations performed by the brute force method. For completeness, we also include the results for a brute force search that uses an early abandon cut-off criterion. The early abandoning is a simple technique that keeps track of the minimal distance found so far, and every time when computing the distance between two new elements, it terminates if it estimates that the current minimum will be exceeded. Note, however, that though the early abandoning can speed up the nearest neighbor search, it still has to perform all pairwise comparisons.

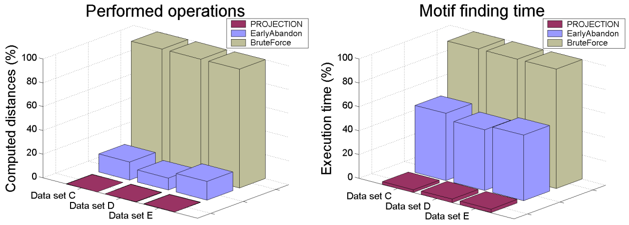


Figure 9: *Brain Activity Data. Left:* Average improvement in distance computation of PROJECTION for a single scaling over the brute force search, and the brute force search extended with early abandon criterion. *Right:* Improvement in running time.

The graph shows that PROJECTION performs approximately 2% of all operations performed by early abandon. This means that the algorithm, prior to the refinement step, has removed from consideration a vast number of the pairwise distance comparisons. The results point out the much better pruning capability of the locality preserving hashing, compared to the popular triangular inequality. For comparison, [7] reports that the triangular inequality prunes between 50%-70% distance computations, which we exceed here notably. This, however, comes at the price of a possibility for some false dismissals.

The speed-up introduced by the method is presented in Figure 9 right. The result does not correlate exactly with the improvement in performed operations, because of the relatively large number of iterations that have to be performed by the algorithm to guarantee that constraint (5) is satisfied (i.e. that the most significant motif will not be omitted). Still, the probabilistic approach takes only 2%-3% of the time necessary for the brute force algorithm to complete.

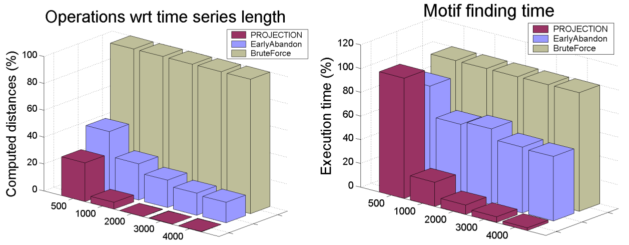


Figure 10: *Brain Activity Data Set C. Left:* Average improvement in performed operations with respect to the time series length. *Right:* Average improvement in running time.

The number of performed operations and the running time naturally dependent on the size of the time series in the data set too. We illustrate this in Figure 10. The figure shows the results for data set C, where for each time series in the training/testing data sets we have taken only the beginning (the first 500, 1000, etc. points). Retraining of the algorithm to obtain the best tuple ($|\Sigma|$, l, $\hat{l}$, I) is required, because the length of the time series determines the number of subsequences that will be extracted from it using a sliding window, which impacts the density of the input space. For shorter time series the second additive term in optimization problem (4) will be naturally smaller, which favors selecting smaller word and projection sizes. For example, when the time series are 500 data points long, the best tuple is $(|\Sigma|, l, \hat{l}, I) = (6, 3, 1, 18)$. For all lengths, the test accuracy of the method remains the same - more than 95% when every single and around 90% when every second scaling step is checked. The worse running time of the algorithm for the data set of lengths 500 is due to the additional implementation overhead for supporting the sparse collision matrix. In our realization, the matrix is implemented as a set of hash tables which we index using the projected SAX words as keys. Incrementing the time series length, however, increases the brute force search requirements quadratically while the PROJECTION solution scales linearly, and thus at some point becomes more efficient.

7.2 Motion-capture Motifs

Motion-capture data finds increasing utility in a number of domains, such as animation, computer games or training simulators [8]. Finding significant motifs under uniform scaling in applications from those areas can also be very useful. For example, detecting the motifs in movements, regardless of slight time stretches, can allow users to interact better with console games, such as the popular Nintendo Wii. In animation, on the other hand, motion-capture sequences are often stitched together to form larger animated episodes. To be more realistic and continuous, the stitching process requires the extrapolation of the first episode with several frames before attaching the second episode. Finding similar motifs and looking at their continuation can help in detecting more suitable extrapolation frames.

In this set of experiments, we have a collection of 75 motion-capture sequences with duration of 10sec-30sec. The motions captured are martial arts movements - kicks, blocks, punches and retracting movements. The time series that we extract from the data comprise the z-coordinate of the sensor attached to the left arm of the actor (see Figure 11). For ease of the evaluation, all time series are resampled to a length of 1200 data points. We split the data into 40 time series training set, and 35 time series test set. For minimum motif length we use $m_q = 120$ which for most sequences corresponds to 2sec-3sec movements.

Similar actions are never repeated by humans in precisely the same way, and rather tend to differ in how they stretch in time. This explains why the Euclidean distance, though robust in general, will fail in these cases. In Figure 11, for example, we search for the best motif under Euclidean distance (the top frames and the graph underneath them) and also under uniform scaling (bottom frames and graph). In the beginning of the sequence the actor repeats the same blocking movement twice but the second occurrence is ∼7% longer than the first one. This is detected by the uniform scaling distance, while the Euclidean distance in this case detects two quite different actions. Note also that the subsequent few frames for the two motif occurrences, detected by the uniform scaling distance, are also quite similar. This demonstrates how very similar motifs can be used in extrapolating motion-capture episodes when building animations.

The data is quite different from the EEG data studied earlier. The time series have no periodicity and no significant noise is present either. The best tuple learned by PROJECTION on the training set is $(|\Sigma|, l, \hat{l}, I) = (4, 5, 5, 1)$. Using projection size $\hat{l}$ equal to the word size l implies that the best motifs for all of the training time series have been symbolized by SAX to the exactly same letter representation. Naturally in this case only one iteration is required to obtain a hit for any of the motifs in the collision table. The accuracy evaluated on the test time series is: 94% (scaling step = 1), 85% (scaling step = 2) and 85% (scaling step = 3). The best motifs were again estimated to require less than 20% scaling. The method again performs less than 10% of the operations performed by early abandon, which is comparable to the results for the EEG series of length 1000.

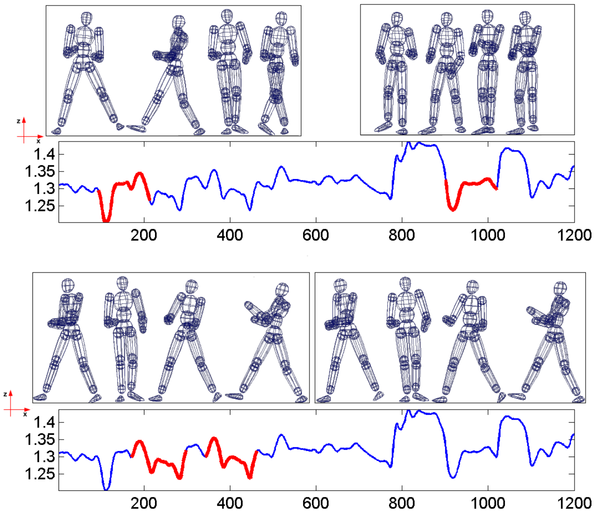


Figure 11: *Motion-capture data. Top:* The best motif detected using Euclidean distance and four corresponding frames extracted from each of the two motif occurrences. *Bottom:* The best motif under uniform scaling. The corresponding frames are part of the same action repeated twice.

7.3 Projectile Shapes

While the ideas in this paper apply only to real-valued time series, it has long been noted that in many cases it is possible to meaningfully convert data types as diverse as DNA [15], text [12], XML, video and shapes [17] into time series. In such cases we believe that our motif finding algorithm may be of utility in those domains. As a concrete example we consider the problem of finding motifs in parts of two-dimensional shapes. Figure 12 shows an example of a projectile point (arrowhead) converted into time series.

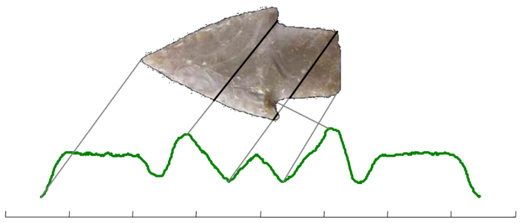


Figure 12: A two-dimensional shape, such as an arrowhead, can be converted to a one-dimensional pseudo time series by tracing the boundary of the shape and recording the local angle.

There are many examples of important databases of shapes in various scientific domains, but as hinted at in Figure 12 we will consider arrowheads here as our motivating domain. At the authors institution, the Lithic Technology Laboratory estimates that they have over one million projectile points, and many other institutions have even larger collections. Figure 13 illustrates the surprising diversity of arrowhead shape.

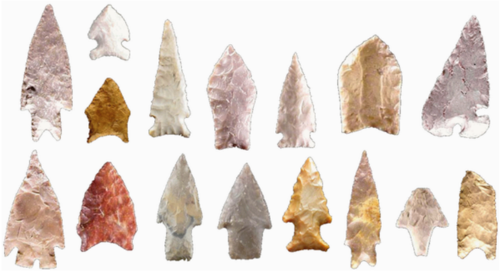


Figure 13: A random selection of arrowheads hints at the great variability of possible shapes. Note that the two in the bottom right corner have broken tips.

We could consider the problem of finding the pair of arrowheads which are most similar, perhaps by hierarchically clustering the shapes and examining the leaf nodes. However, anthropologists are typically more interested in the arrangement of local details [25] (barbs, tips, shoulders, tangs etc). In addition, many arrowheads are broken, frustrating attempts at whole matching.

A query suggested to us by an anthropologist is to examine a large collection of arrowheads to find smaller reoccurring details. We formalized this by assuming the region of interest occupied one quarter of the boundary.

Note that the one-dimensional representation of shapes we are using does not guarantee rotation invariance, we solve this problem simply by concatenating each signal with one quarter of itself as shown in Figure 14.

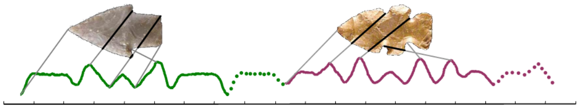


Figure 14: Converting a collection of shapes to a format suitable for motif mining. Note that because we are interested in motifs which occupy about 1/4 of the boundary, we have concatenated 1/4 of each signal to itself (dashed line).

Given this representation there is only one minor modification to be made to our algorithm. As the sliding window is moving across the long time series which represents the entire collection, it will not extract subsequences which contain elements from two different shapes (i.e two different colors in Figure 14).

We performed an experiment on a database of 1,231 diverse arrowheads which come from all over North America. These images were obtained from the UCR Lithic Technology Laboratory and from various public domain resources.

Motif discovery without uniform scaling revealed the obvious but uninteresting motif of arrowhead tips. However, when we attempted motif discovery allowing uniform scaling of up to 40%, several interesting motifs did occur. For brevity we will just consider the best motif, which is shown in Figure 15.

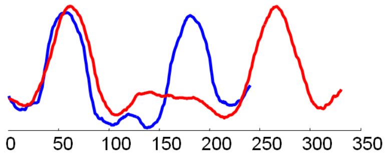


Figure 15: The best motif discovered under a maximum uniform scaling of 40%. The shorter sequence closely matches the longer one when scaled up by 38%.

Figure 16 shows the motifs in context in the original arrowheads.

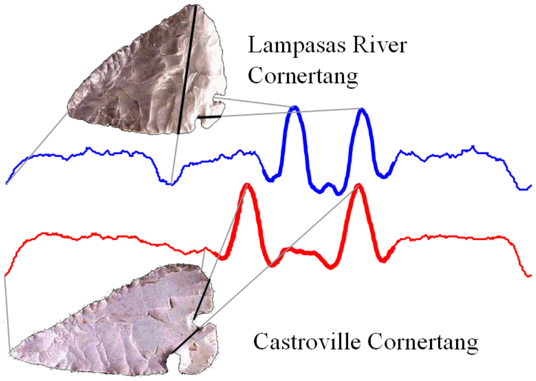


Figure 16: The best motif under uniform scaling corresponds to the double notches of cornertang arrowheads.

In our follow up investigation we found that cornertang shape has long intrigued anthropologists. These objects are relatively rare and are found almost exclusively in Texas.

It is important to note that the shapes of the two full arrowheads shown in Figure 16 are not particularly close when measured under the Euclidean, Warping, Chamfer or Hausdorff [17] distance measures (The Castroville cornertang is much longer and more pointed). It is only by examining subsections that local similarities are revealed.

8. CONCLUSIONS

We studied the problem of detecting time series motifs, which occur with different stretch along the time axis. Such motifs were demonstrated to have important utility in areas as diverse as medical recording analysis, improving game interactivity and animation, or in categorization of shapes. The work introduced an effective and efficient approach for identifying the existing motifs. The algorithm learns a suitable parametrization in a fully unsupervised manner, using a training set of time series from the domain under study. Though probabilistic, the scheme was shown to have a very low rate of omitting the true motifs.

Our current efforts are targeted towards applying this methodology as part of other learning tasks as subsequence clustering and classification, novelty detection, and even forecasting. We are also exploring the applicability of the approach to other applications, such as the categorization of songs by only using small representative tunes.

9. ACKNOWLEDGMENTS

We would like to thank Dr. Leslie A. Quintero and Dr. Philip J. Wilke of the UCR Lithic Technology Laboratory, for providing us with the projectile samples used in the current evaluation.

10. REFERENCES

[1] H. Abe and T. Yamaguchi. Implementing an integrated time-series data mining environment - a case study of medical kdd on chronic hepatitis-. *1st International Conference on Complex Medical Engineering (CME2005)*, 2005.

[2] I. Androulakis. New approaches for representing, analyzing and visualizing complex kinetic mechanisms. In *Proc. of the 15th European Symposium on Computer Aided Process Engineering*, 2005.

[3] I. Androulakis, J. Wu, J. Vitolo, and C. Roth. Selecting maximally informative genes to enable temporal expression profiling analysis. In *Proc. of Foundations of Systems Biology in Engineering*, 2005.

[4] R. Andrzejak, K. Lehnertz, F. Mormann, C. R. P. David, and C. Elger. Indications of nonlinear deterministic and finite-dimensional structures in time series of brain electrical activity: dependence on recording region and brain state. *Phys Rev E Stat Nonlin Soft Matter Phys*, 64, 2001.

[5] D. Arita, H. Yoshimatsu, and R. Taniguchi. Frequent motion pattern extraction for motion recognition in real-time human proxy. In *Proc. of JSAI Workshop on Conversational Informatics*, pages 25–30, 2005.

[6] J. Buhler and M. Tompa. Finding motifs using random projections. In *Proc. of the 5th International Conference on Computational Biology*, pages 69–76, 2001.

[7] W. Burkhard and R. Keller. Some approaches to best-match file searching. *Commun. ACM*, 16(4):230–236, 1973.

[8] B. Celly and V. Zordan. Animated people textures. In *Proc. of 17th International Conference on Computer Animation and Social Agents (CASA)*, 2004.

[9] B. Chiu, E. Keogh, and S. Lonardi. Probabilistic discovery of time series motifs. In *Proc. of the 9th International Conference on Knowledge Discovery and Data mining (KDD'03)*, pages 493–498, 2003.

[10] F. Duchêne, C. Garbay, and V. Rialle. Apprentissage non supervise de motifs temporels, multidimensionnels et hétérogènes. application a la télésurveillance médicale. *CAP*, 2005.

[11] R. Duda, P. Hart, and D. Stork. *Pattern Classification*. Wiley-Interscience, 2000.

[12] P. Fung. A pattern matching method for finding noun and proper noun translations from noisy parallel corpora. In *Proc. of the 33rd annual meeting on Association for Computational Linguistics*, pages 236–243, 1995.

[13] R. Hamid, S. Maddi, A. Johnson, A. Bobick, I. Essa, and C. Isbell. Unsupervised activity discovery and characterization from event-streams. In *Proc. of the 21st Conference on Uncertainty in Artificial Intelligence (UAI05)*, 2005.

[14] P. Indyk, R. Motwani, P. Raghavan, and S. Vempala. Locality-preserving hashing in multidimensional spaces. In *Proc. of the 29th ACM symposium on Theory of computing*, pages 618–625, 1997.

[15] E. Keogh, S. Lonardi, V. Zordan, S. Lee, and M. Jara. Visualizing the similarity of human and chimp DNA. *Multimedia Video, http://www.cs.ucr.edu/ eamonn/DNA/*, 2005.

[16] E. Keogh, T. Palpanas, V. Zordan, D. Gunopulos, and M. Cardle. Indexing large human-motion databases. In *Proc. of the 30th International Conference on Very Large Data Bases (VLDB04)*, pages 780–791, 2004.

[17] E. Keogh, L. Wei, X. Xi, S. Lee, and M. Vlachos. LB_Keogh supports exact indexing of shapes under rotation invariance with arbitrary representations and distance measures. *VLDB*, 2006.

[18] E. J. Keogh. Efficiently finding arbitrarily scaled patterns in massive time series databases. In *PKDD*, pages 253–265, 2003.

[19] J. Lin, E. Keogh, S. Lonardi, and P. Patel. Finding motifs in time series. In *Proc. 2nd Workshop on Temporal Data Mining (KDD'02)*, 2002.

[20] J. Maizel and R. Lenk. Enhanced graphic matrix analysis of nucleic acid and protein sequences. In *Proc. Natl. Acad. Sci. USA*, volume 78, pages 7665–7669, 1981.

[21] A. McGovern, D. Rosendahl, A. Kruger, M. Beaton, R. Brown, and K. Droegemeier. Understanding the formation of tornadoes through data mining. *5th Conference on Artificial Intelligence and its Applications to Environmental Sciences at the American Meteorological Society*, 2007.

[22] D. Minnen, T. Starner, I. Essa, and C. Isbell. Discovering characteristic actions from on-body sensor data. *10th International Symposium on Wearable Computers (ISWC06)*, 2006.

[23] K. Murakami, S. Doki, S. Okuma, and Y. Yano. A study of extraction method of motion patterns observed frequently from time-series posture data. *IEEE International Conference on Systems, Man and Cybernetics*, 4:3610–3615, 2005.

[24] P. Pevzner and S. Sze. Combinatorial approaches to finding subtle signals in DNA sequences. In *Proc. of the 8th International Conference on Intelligent Systems for Molecular Biology*, pages 269–278, 2000.

[25] D. Rogers. Corner tang knives across Texas. *Personal Press*, 2000.

[26] S. Rombo and G. Terracina. Discovering representative models in large time series databases. In *Proc. of the 6th International Conference On Flexible Query Answering Systems*, pages 84–97, 2004.

[27] T. Sauer. Time series prediction by using delay coordinate embedding. *Time Series Prediction. Forecasting the Future and Understanding the Past*, 59(8):175–193, August 1994.

[28] Y. Tanaka, K. Iwamoto, and K. Uehara. Discovery of time-series motif from multi-dimensional data based on MDL principle. *Mach. Learn.*, 58(2-3):269–300, 2005.