Time Series Motif Discovery: A Comprehensive Evaluation

Valerio Guerrini
Univ. Paris Saclay, Univ. Paris Cité, CNRS,
SSA, ENS Paris Saclay, INSERM, Centre Borelli
valerio.guerrini@ens-paris-saclay.fr

Thibaut Germain
Univ. Paris Saclay, Univ. Paris Cité, CNRS,
SSA, ENS Paris Saclay, INSERM, Centre Borelli
thibaut.germain@ens-paris-saclay.fr

Charles Truong
Univ. Paris Saclay, Univ. Paris Cité, CNRS,
SSA, ENS Paris Saclay, INSERM, Centre Borelli
charles.truong@ens-paris-saclay.fr

Laurent Oudre
Univ. Paris Saclay, Univ. Paris Cité, CNRS,
SSA, ENS Paris Saclay, INSERM, Centre Borelli
laurent.oudre@ens-paris-saclay.fr

Paul Boniol
Inria, ENS, DIENS
CNRS, PSL
paul.boniol@inria.fr

ABSTRACT

Motif Discovery involves identifying recurring patterns and locating their occurrences within a time series without prior knowledge about their shape or location. In practice, Motif Discovery faces several data-related challenges, leading to various definitions of the problem and multiple algorithms addressing these challenges to different extents. However, there has been no systematic evaluation and comparison of these diverse approaches. Consequently, this paper presents a comprehensive literature review covering data-related challenges, motif definitions, and algorithms. We also analyze the strengths and limitations of algorithms carefully chosen to represent the literature diversity. The analysis is structured around key research questions identified from our review. Our experimental findings provide practical guidelines for selecting Motif Discovery algorithms suitable for a given task and suggest directions for future research.

PVLDB Reference Format:
Valerio Guerrini, Thibaut Germain, Charles Truong, Laurent Oudre, and Paul Boniol. Time Series Motif Discovery: A Comprehensive Evaluation. PVLDB, 18(7): 2226 - 2239, 2025.
doi:10.14778/3734839.3734857
PVLDB Artifact Availability:
The source code, data, and/or other artifacts have been made available at https://github.com/grrvlr/TSMD.

1 INTRODUCTION

Time Series are prevalent in many scientific and industrial domains [13, 56]. Formally, a time series is a sequence of time-ordered real-valued samples. The samples can correspond to different physical quantities, such as temperature and pressure [6], electricity consumption [59], or human pose [13]. Several tasks have emerged from the growing desire to analyze time series like classification [4], clustering [57], anomaly detection [58, 68], and Motif Discovery [10, 67]. The latter involves discovering recurring patterns, known as motifs, within a time series. Beneficial for explanatory purposes, this task can also be a first step toward subsequent analysis. For instance, classification or clustering of long-time series may be impractical but can be simplified by extracting representative features, such as motifs. Such motifs usually represent temporal events, such as heartbeats in an electrocardiogram [23] or the electrical consumption of an appliance in a smart-meter series [59].

Proceedings of the VLDB Endowment, Vol. 18, No. 7 ISSN 2150-8097.
doi:10.14778/3734839.3734857

In practice, time series raises several challenges when performing Motif Discovery. For instance, occurrences (repetitions) of a given motif may suffer from deformations like amplitude scaling or offset shift, which must be considered when comparing occurrences. Also, the length of occurrences may vary depending on their time parametrization, and dealing with such time-warping deformations is challenging. Moreover, a single time series may contain several motifs of different lengths, stressing the need for methods for discovering motifs at various time scales.

Over the past two decades, researchers have proposed various definitions of Motif Discovery problems and derived several algorithms to address these challenges to some extent [2, 10, 17, 18, 22, 35–37, 40, 64, 67, 69, 80, 84, 87, 88]. However, comparing algorithms developed from different problem formulations is challenging, even when they share a common objective. As a result, despite substantial academic interest, there has been no systematic evaluation and comparison of these diverse approaches. Consequently, selecting the most suitable algorithm for a specific Motif Discovery task remains an open question.

To tackle the limitation mentioned above, this paper first provides an exhaustive and comprehensive review of the literature regarding the Motif Discovery problem's definition, algorithms, and data-related challenges. This study only focuses on methods dealing with real-valued univariate time series, resulting in a pool of 55 methods classified into three different families. We also present a comprehensive comparison and evaluation of a representative set of 11 Motif Discovery algorithms. Algorithms' performances, in terms of f1score and time efficiency, are evaluated on 8 real-world time series datasets and several synthetic datasets specifically designed to measure algorithms' robustness to challenges commonly encountered with temporal data. Experimental results provide practical guidelines for selecting suitable Motif Discovery algorithms for a given task and suggest directions for future works. Overall, our contributions are as follows:

- We thoroughly present the literature on the time series Motif Discovery problem by discussing its historical evolution and trends (Section 2.1), enumerating several subproblem formulations (Section 2.3) belonging to two generic problems.
- We propose a process-centric taxonomy in which the presented Motif Discovery algorithms are grouped under three families. We also provide additional characteristics for each method, such as its preprocessing, the similarity measure used, and if it handles motifs' length variabilities (Section 2.4).
- We enumerate different challenges encountered in real-world applications and derive six research questions relevant to the Motif Discovery task raised by these challenges (Section 2.5).
- We introduce our experimental benchmark (Section 3), which includes 11 Motif Discovery algorithms, 8 real-world labeled

datasets (569 time series in total), a synthetic data generator, and performance metrics.
- We present our experimental evaluation to address 6 research questions motivated by data-related challenges (Section 4). Specifically, we compared algorithms' f1scores and execution time on real-world datasets to evaluate performances on various applications. The remaining questions address algorithms' strengths and limitations on synthetic data generated from specific real-world inspired scenarios.
- We propose guidelines allowing users to choose which method to use depending on the characteristics of the time series.
- We provide our material, including methods implementations, datasets, and experiments, in an open-source repository [1].

We conclude this paper with a discussion of the implications of our work and future directions in Motif Discovery for time series.

2 TIME SERIES MOTIF DISCOVERY

This section first clarifies mathematical definitions and notations related to time series Motif Discovery. It also provides a historical review of research advances in this area. We then propose a novel taxonomy for Motif Discovery algorithms, and we enumerate the practical data-related challenges. These challenges raise research questions presented and addressed in Section 4.

2.1 Motif Discovery: A brief History

Motif Discovery in real-valued time series implies identifying similar subsequences (representing a specific pattern or motif) within the time series [39]. Before the formal introduction of Motif Discovery [39, 55], most of the research studies focused on the problem of identifying already known patterns (also referred to as query by content) [1, 11, 20, 28–30, 71], or in identifying patterns (or motifs) in discrete time series, in particular in computational biology [7, 15, 25, 60, 70], which had a strong influence in the early real-valued time series Motif Discovery literature.

Indeed, inspired by the work in computational biology, Motif Discovery in real-valued time series has been first tackled by proposing methods with a preprocessing step of discretizing time series [12, 16, 39, 43, 44, 62, 73, 74, 82]. These methods directly search for patterns in the discretized series, which reduces computation times but only provides approximated solutions. Toward exact results in a reasonable time, many papers have focused on the simplified but well-posed problem named Best Motif Pair, which consists of finding the subsequence pairs with minimum distances under some non-overlapping conditions [2, 10, 18, 33, 34, 40, 45–49, 53, 54, 77, 78, 82, 84, 87, 88].

Overall, in the two last decades, we observe several key moments in the resolution of the Motif Discovery problem: (i) The introduction of the general problem in 2002 [39], (ii) the proposal of the first algorithm to solve the Motif Pair problem exactly in 2009 [49], (iii) the use of grammar-based techniques on the discretized time series in 2010 [17, 18, 35, 36, 69], and (iv) the introduction of Matrix Profile in 2016 [84], which resulted in significantly reducing execution time and on which many of the latest Motif Discovery algorithms are based [2, 37, 40, 87, 88].

More recently, several Motif Discovery methods [22, 67, 80] were introduced to solve the Motif Set problem. These methods aim to maintain reasonable scalability without using a discretization step, which is potentially difficult to calibrate. Finally, it is interesting to note the recent emergence of auto-encoder methods for Motif Discovery [5, 52, 63]. However, unlike all the other methods mentioned above, a training phase is necessary, limiting their applications to use cases with a large set of time series at our disposal.

[1]Public repository: https://github.com/grrvlr/TSMD

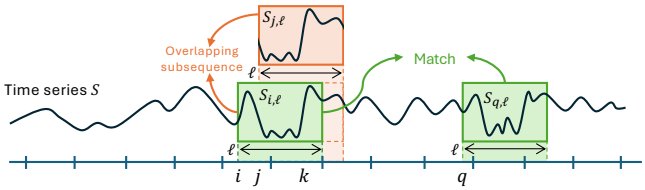


Figure 1: Illustration of match, and overlapping subsequences

2.2 Time Series and Motifs Notations

Although there exist methods to solve Motif Discovery on symbolic [7, 72] or multivariate time series [42, 83], most of the methods in the literature address univariate real-valued case, and in what follows, we only discuss this case. We now introduce fundamental definitions to assess the technical differences between the problem formulations and the proposed algorithms.

Definition 1 (Univariate real-valued time series). *An univariate real-valued time series of length n is a time-ordered sequence $S = [s_1, \ldots, s_n]$ of n coefficients in $\mathbb{R}$.*

In the following, we refer to univariate real-valued time series and time series without distinction. We first start by defining formally the concept of subsequence:

Definition 2 (Subsequence). *The subsequence of a time series $S \in \mathbb{R}^n$ of length ℓ and starting at index $i \in [1, \ldots, n-\ell+1]$ is the sequence $S_{i,\ell} = [s_i, \ldots, s_{i+\ell-1}]$.*

For example, $S_{i,l}$, $S_{j,l}$ and $S_{q,l}$ illustrated in Figure 1 are subsequences of S. We now define the concept of matching subsequences:

Definition 3 (Match). *Given a threshold $R > 0$, the subsequences $S_{i,\ell}$ and $S_{j,\ell}$ of a time series $S \in \mathbb{R}^n$ are matching if and only if $d(S_{i,\ell}, S_{j,\ell}) < R$.*

For example, $S_{i,l}$ and $S_{q,l}$ in green in Figure 1 are matching. However, a difficulty encountered in the Motif Discovery task [31] is the following: for almost every subsequence of a time series S, the best match will be the subsequence just before or after the one considered. The notion of overlapping subsequences was introduced to cope with this limitation, and formally defined as follows:

Definition 4 (Overlapping subsequences). *Two subsequences $(S_{i,\ell}, S_{j,\ell'})$ of a time series $S \in \mathbb{R}^n$ with $i < j$ overlap if $j \leq i + l$.*

In Figure 1, $S_{i,l}$ and $S_{j,l}$ (in green and orange) are overlapping subsequences. Based on these definitions, we can now examine the formal problems of Motif Discovery introduced in the literature.

2.3 Motif discovery: A multifaceted Problem

Attesting to the challenging nature of the problem, we observe several definitions of the Motif Discovery task. Indeed, if we consider the vague definition of motifs as a set of subsequences of a time series fairly close to each other, the interpretation of *fairly close* can lead to very different definitions of motifs. To assess the variety of definitions and, therefore, the potential ambiguity of the

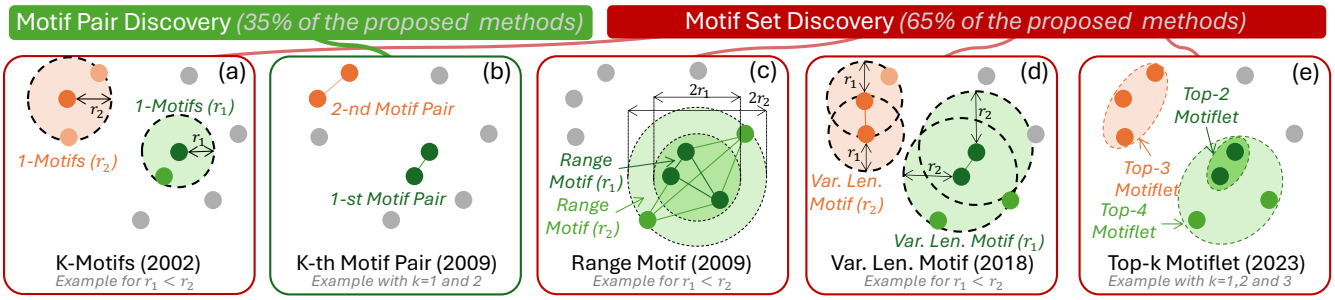


Figure 2: Motif Discovery problems proposed in the literature inspired by [67]

problem, we provide below a historical, non-exhaustive list of the main definitions in the literature.

- *K*-**Motifs** (2002) [39] (Figure 2(a)): Given a time series S, a subsequence length ℓ and a range R, the most significant motif in S (called 1-Motif) is the subsequence C_1 that has the highest count of non-overlapping matches (ties are broken by choosing the motif whose matches have the lower variance). The kth most significant motif in T (called K-Motif) is the subsequence C_k that has the highest count of non-overlapping matches and satisfies $D(C_K, C_i) > 2R$, for all $1 \leq i < K$.
- *k*-**th Motif Pair** (2009) [49] (Figure 2(b)): The Best Pair Motif of length ℓ of a time series $S \in \mathbb{R}$ is the unordered pair of time series subsequences $S_{i,l}$, $S_{j,l}$ of S which is the most similar among all possible non-overlapping pairs. The kth-Pair Motif of length ℓ of a time series $S \in \mathbb{R}^n$ is the kth most similar non-overlapping pair of subsequences of S.
- **Range Motif** (2009) [49] (Figure 2(c)): The Range Motif with range r is the maximal set of time series subsequences such that the maximum distance between them is less than $2r$.
- **Variable Length Motif** (2018) [37] (Figure 2(d)): Let $\{S_{\alpha,\ell}, S_{\beta,\ell}\}$ be a Motif Pair of length ℓ of data series $S \in \mathbb{R}^n$. The Motif Set $\mathcal{M}_{r,\ell}$ is: $\mathcal{M}_{r,\ell} = \{S_{i,\ell} \mid \text{dist}(S_{i,\ell}, S_{\alpha,l}) < r \text{ or } \text{dist}(S_{i,l}, S_{\beta,l}) < r\}$
- **Top-***k* **Motiflet** (2023) [67] (Figure 2(e)): Given a time series S, cardinality $k \in \mathbb{N}$ and length ℓ, the top k-Motiflet of S is the set $\mathcal{M}$ with $|\mathcal{M}| = k$ subsequences of S of length ℓ with minimal extent. Where the extent of a set $\mathcal{M}$ is the maximal pairwise distance between subsequences of $\mathcal{M}$.

We could complete this list with many variants of the examples above (*K-Motif*(n, R, d) [12], *k-ball* [38], *Latent Motif* [24], *Uniform Scaling Motif* [82]). This vast list of problem definitions shows the ambiguity of Motif Discovery and the difficulty of providing a unique benchmark. However, we can distinguish between two prominent families of problems, classified according to the nature of the object returned by the methods. The first abstract problem formulation is as follows:

Problem 1 (**Pair Motif Discovery**). *Identifying the two most similar non-overlapping subsequences in a time series.*

Even though Problem 1 only encapsulate *K-th Motif Pair* problems, it concerns approximately more than 35% of the methods proposed in the literature (c.f. Figure 2 and Figure 3). In addition, Problem 1 is well-posed, and multiple exact and approximate methods with moderate complexity exist [48, 49, 84]. However, this definition does not align with real-world applications where users seek all occurrences of the desired patterns. For example, finding only the most similar pair in electrical consumption time series is insufficient for many applications, such as unsupervised appliance detection for electrical consumption prediction [59]. In practice, it can be valuable for practitioners to have methods that provide a complete set of subsequences corresponding to a given motif. Toward that direction, we can state the problem as follows:

Problem 2 (**Motif Set Discovery**). *Identifying sets of subsequences that encompass every occurrence of distinct repeated patterns in a time series.*

While Problem 2 is deliberately more abstract than Problem 1, encompassing multiple formal definitions, it is also more general and better aligned with real-world applications of Motif Discovery. Moreover, it is important to note that the *Pair Motif Discovery* Problem can be seen as a sub-problem of the *Motif Set Discovery* Problem. Once the motif pair has been found, a Motif Set can be built around it. Several methods have been proposed in the literature to post-process the output of *Pair Motif* methods to solve the *Motif Set* problem [3, 50]. For example, VALMOD [37], which initially solves the *Pair Motif Discovery* problem, can then build around the identified motifs pair a set of motifs (i.e., solving the *Motif Set Discovery* problem).

Therefore, we can evaluate and compare all the Motif Discovery methods proposed in the literature, regardless of the problem they initially solve, within the general problem of *Motif Sets Discovery*.

2.4 Process-centric Taxonomy

As mentioned above, Motif Discovery has attracted considerable academic attention, and various methods have been proposed. However, as mentioned in Section 2.3, the methods proposed in the literature differ in terms of the definition of motif considered. In addition, we also observe significant differences in terms of methodology and challenges they attempt to address. Therefore, we propose a process-centric taxonomy of Motif Discovery methods. Overall, we highlight three prominent families: (i) *Frequency-based*, (ii) *Similarity-based*, and (iii) *Encoding-based*. We depict our taxonomy in Figure 3 and detail each family below.

2.4.1 ***Frequency-based Methods***. *The frequency-based methods aim at identifying sets of subsequences that represent the most frequently repeated patterns.* Many methods in this family take a radius R as a parameter and try to find the set of non-overlapping subsequences of maximum cardinality that fit within a circle of radius R [3, 12, 24, 38, 39, 44]. Some methods with a prior discretization step make use of the fact that in a discretized time series, the exact repetitions of an element can be counted to return the exactly repeating elements with the highest cardinality [8, 85].

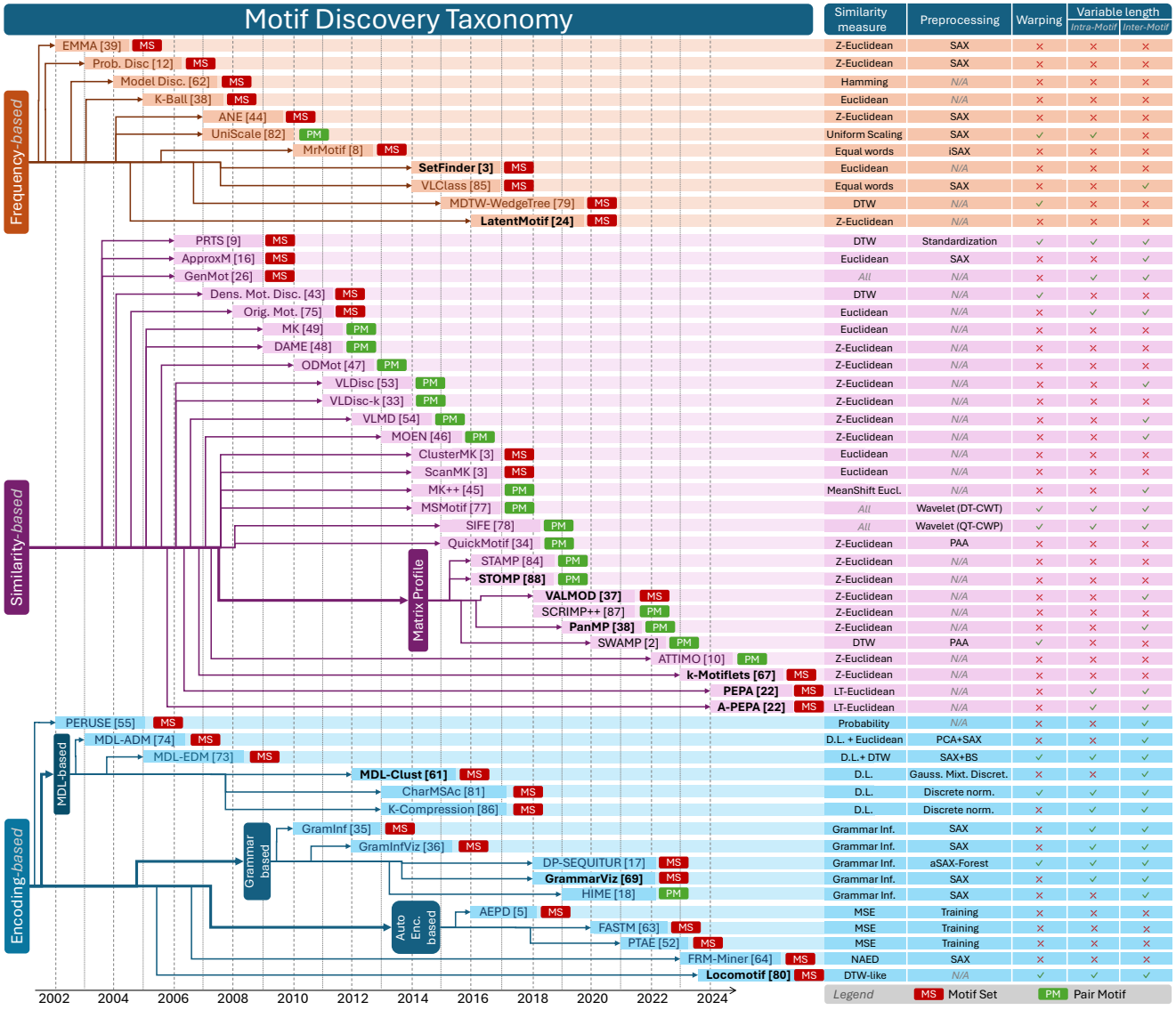


Figure 3: Detailed Motif Discovery Taxonomy. DL:Description Length, MDL: Minimum Description Length, MSE: Mean Square Error, NAED: Normalized Average Euclidean Distance, PAA: Piecewise Aggregate Approximation, SAX: Symbolic Aggregate approXimation

2.4.2 ***Similarity-based Methods.*** *The Similarity-based methods aim at identifying a set of subsequences with minimum distance between occurrences independently of the number of occurrences in the set.* This is particularly true for all methods that rank the relevance of a Motif based on the minimum pairwise distance between two occurrences. Pair Motif is the extreme case where Motif Sets are composed only of the two closer nonoverlapping subsequences. Building on Pair Motifs algorithms, some Motif Set algorithms first find Pair Motifs and construct the Motif Set around the Pair Motifs [3, 37]. Motifs can also be ranked based on other proximity criteria, such as the maximum pairwise distance of a set of occurrences given a fixed number of occurrences [67].

2.4.3 ***Encoding-based Methods.*** *The Encoding-based algorithms aim at identifying a set of subsequences that represent the best way to encode the time series according to different criteria.* In the literature, we find 3 main ways of encoding time series to find motifs: by using information theory and the Minimum Description Length (MDL) principle to rank the capacity of motifs to encode a time series [73, 74, 81], by using Grammar Inference algorithms in which motifs are seen as grammar rules and their occurrences as instances of the rules [17, 18, 35, 36, 69], by training AutoEncoders to reconstruct the time series with a minimum number of motifs.[5, 52, 63]

2.4.4 ***Discussion of the literature.*** Figure 3 highlights several interesting findings. First, the most recently proposed methods belong to *Similarity-based* and *Encoding-based* families. Second, even though the first methods proposed to solve the *Motif Pair* problem belongs to the *Frequency-based* family, most of the remaining methods belong to the *Similarity-based* family (60% of the methods belonging to this family aims to solve the *Motif Pair* problem, compared to only one for each of the two other families). Lastly, a

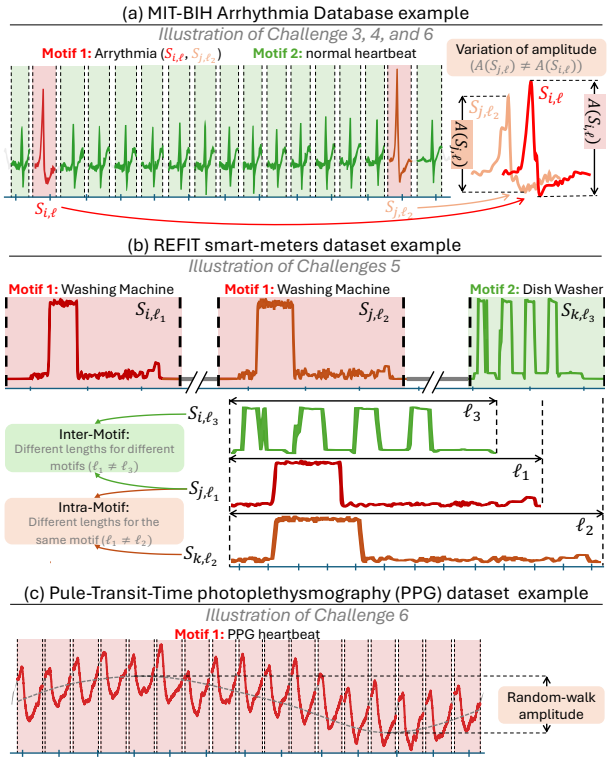


Figure 4: Illustrations of Motif Discovery practical challenges with real datasets examples.

large majority of the methods in the *Encoding-based* claim to detect variable length motifs, either intra-motifs (i.e., variable length occurrences of the same motifs) or inter-motif (i.e., different motifs with different lengths). This can be explained by the fact that these methods are not necessarily based on distance measurements that impose rigid motif structures.

Based on this taxonomy, we select for our experimental evaluation a subset of methods that cover most of the categories introduced in the literature (highlighted in bold in Figure 3).

2.5 Motif Discovery in Practice

Beyond the different problem formulations and the different families of methods, the literature is structured around several practical challenges faced when applying methods to real-world data (Figure 4). In theory, the proposed methods have addressed these challenges in different ways. However, there has never been a comprehensive study comparing the methods in terms of their practical effectiveness in addressing these challenges. Below, we list the most important Motif Discovery challenges relevant to real-world applications and some hints on how they have been dealt with in theory.

Challenge 1: Performances on real data. Practitionners generally want to find patterns representing interpretable temporal events in real-life applications. The first challenge is, therefore, to have a model that is generic enough to detect such temporal events, which is characterized by good performance on expert-labeled data.

Challenge 2: Scalability for long time series. The Motif Discovery task is computationally expensive. Indeed, the time complexities of the methods are generally quadratic in the length of the time series. A significant challenge lies in the scalability of the algorithms. This problem has been addressed in different ways, for example, by solving computationally tenable subproblems such as the Pair Motif Problem [48, 49, 84].

Challenge 3: Presence of several different motifs. Time series can either contain one or several different motifs. The presence of a large number of motifs is a major challenge, as the methods must not only detect the occurrences, but also group them together. This problem has been dealt with by searching for the K best motifs [3, 22, 37, 61] and not only the best one. The number of motifs can be a user given parameter [3, 37] or found by some heuristics [22, 61].

Challenge 4: Motifs cardinality. The cardinality of a motif refers to its number of occurrences. Motifs can have variable cardinalities and, therefore, be more or less rare. Moreover, there may also be imbalances between the cardinalities of different patterns when there are several motifs in the time series. The cardinality of patterns represents a real challenge since some approaches prioritize the most frequent patterns [3, 24], while others prioritize patterns containing the closest occurrences [37, 84], independently of cardinality.

Challenge 5: Variable length Variations in motif length pose a significant challenge in time series analysis. These variations can occur within the same motif due to deformations, expansions, or contractions, a phenomenon known as time warping. This issue is commonly addressed using elastic distance measures based on Dynamic Time Warping (DTW) [2, 9, 14, 42, 73, 77, 79, 80]. Other approaches allow motifs to have different lengths by merging overlapping occurrences [22, 26, 75] or by collapsing successive identical symbols in discretized time series [35, 36, 69]. Length variations can also result from the presence of multiple motifs with different average lengths, known as inter-motif variability. For example, in electrical consumption time series [59], motifs can appear on an hourly or daily scale. This variability complicates the comparison of motifs relevance when they have different average lengths. To address this, several methods propose heuristics to determine optimal motif lengths [43, 73, 74, 86], use grammar rules without predefined lengths [17, 18, 35, 36, 69, 85], or enumerate motif occurrences over a range of window lengths for comparison [37, 40, 45, 46, 53, 54].

Challenge 6: Spatial deformations. Occurrences of the same motif can be affected by deformations such as amplitude scaling, offset shifts, linear trends, and noise. These variations pose a challenge, as classical distances like Euclidean distance do not account for all these factors. Z-normalized Euclidean distance was introduced to handle amplitude variations and offset shifts by normalizing subsequences' standard deviations and means, effectively solving this issue. However, linear trends have received little attention in the literature. A recent advancement, LT-normalized Euclidean distance [21], extends Z-normalization to address this challenge. Noise is typically managed implicitly through distance and similarity measures that allow some variability in the motifs sets.

The taxonomy presented in the previous section gives us an overview of the methods families and indications of which challenges are theoretically addressed by which methods. However, to our knowledge, no benchmark or extensive experimental evaluation exists to compare the performances of the proposed methods. In this paper, we propose to carry out this evaluation by looking at the problem through the prism of the generic Motif Set Discovery problem as defined in Problem 2, and relying on 6 research questions (noted RQ1 to RQ6 and addressed in Section 4) arising directly from the challenges that practitioners may encounter.

Table 1: Our proposed collection of labeled time series. *Ratio* indicates the time series percentage corresponding to a motif. i.M stands for Intra-Motif (length), and I.M. stands for Inter-Motif (length).

Dataset	# TS	TS len.	# motifs /# per TS /ratio	avg. motif len.	i.M (std)	I.M.. (std)
arm-CODA [13]	64	8,050	7/5/0.65	520	22	88
mitdb [23]	100	20,000	10/1.6/0.99	281	36	10
mitdb1 [23]	100	20,000	1/1/0.98	320	12	0
ptt-ppg [41]	100	20,000	1/1/0.98	324	15	0
REFIT [51]	100	210,860	3/2.2/0.08	410	11	34
SIGN [27]	50	173,300	3/3/0.10	57	7	2
JIGSAWMaster [19]	23	10,300	8/3.8/0.66	156	38	66
JIGSAWSlave [19]	32	10,160	9/3.9/0.65	146	35	60

3 PROPOSED BENCHMARK

In order to address the research questions enumerated in the previous section, we have to carefully select algorithms from different families to best represent the diversity of Motif Discovery techniques. In addition, a broad range of domains and applications should be represented in our labeled time series collection. In the following section, we first describe in detail our proposed benchmark, composed of (i) 8 annotated time series datasets and (ii) 11 Motif Discovery algorithms. We summarize our benchmark in Table 1 for the datasets and Table 2 for the methods. Finally, to ease replication and re-usability, we provide an open-access (c.f. Artifact Availability) to our datasets and methods implementations.

3.1 Real Time Series Collection

This section presents the different real datasets used to evaluate Motif Discovery methods. Our selection criteria are the following: (i) The datasets should cover a large scope of time series type, and (ii) the selected time series should highlight the enumerated challenges in Section 2. Our selected datasets are the following:

arm-CODA [13]: is a dataset of 240 multivariate time series collected using 34 Cartesian Optoelectronic Dynamic Anthropometers (CODA) placed on the upper limbs of 16 healthy subjects, each of whom performed 15 predefined movements. Each sensor records its position in 3D space. To construct the dataset, we kept the left forearm sensor of ID 29 and 5 of the predefined movements. The occurrences of the five movements were randomly placed along the time axis for each subject, sensor, and dimension. Gaussian noise with a signal-to-noise ratio of 0.01 is added to all time series. This resulted in a dataset of 64 univariate time series.

mitdb1 [23]: The MIT-BIH Arrhythmia Database contains 48 half-hour recordings of two-channel ambulatory electrocardiograms (ECGs) sampled at 360Hz. Cardiologists annotated the heartbeats according to 19 categories. We divide all recordings into a time series of 1 minute and keep only the first channel. We selected time series of healthy subjects (according to [65]) that contains only normal heartbeats and randomly selected 100 time series.

mitdb [23]: We randomly selected 100 one-minute time series from the MIT-BIH dataset (healthy subjects or not). Each time series has 1 to 4 motifs (normal heartbeats and different types of arrhythmia), each with several occurrences.

ptt-ppg [41]: Pule-Transit-Time photoplethysmogram (PPG) consists of time series recorded with sensors (sampled at 500Hz) from healthy subjects performing physical activities. The annotated motifs are heartbeats. We randomly select 100 40-second-long signals from the first channel of the PPG during the "run" activity.

REFIT [51]: This dataset provides aggregate and individual appliance load curves at 8-second sampling intervals from 20 houses. We selected 10 houses and aggregated recordings of the appliances: dishwasher, washing machine, and tumble dryer. The recordings were down-sampled to 32-second intervals and divided into time series of one week. We kept 10 time series for each house in which the appliances were not used simultaneously. It resulted in a 100 univariate time series dataset with a maximum of 3 different motifs.

SIGN [27]: This dataset is built from samples of Australian sign language. 95 signs were collected from five signers, totaling 6650 sign samples. Based on this, we generate a long time series by injecting several words (concatenation of signs). The different injected signs are the motifs. Every word is separated with flat sequences (i.e., without any motifs). In total, we generate 50 different time series.

JIGSAW [19]: This dataset contains time series recorded from the DaVinci Surgical System. Each time series contains 76 dimensions (i.e., sensors) with an acquisition rate of 30 Hz. The sensors are divided into two groups: patient-side manipulators (**JIGSAWSlave**), and master tool manipulators (**JIGSAWMaster**). The recorded time series corresponds to surgeons performing a suture that can be decomposed into 11 gestures. Each gesture corresponds to a motif that can be repeated multiple times within the same time series. Overall, we selected 23 time series (from different sensors) for JIGSAWMaster and 32 time series for JIGSAWSlave.

3.2 Synthetic generator

This section presents the synthetic time series generator used to perform the experiments corresponding to RQ 2 to 6.

For a given number of motifs K, the generator constructs one representative per motif. Given an average length l_i, and a fundamental frequency (set to 4Hz in our case), a motif representative is generated as the sum of the sine function of the l_i first harmonics, with the phases and the amplitudes uniformly sampled over $[\pi, \pi]$ and $[1, 1]$. The k_i occurrences of motif i are then constructed by temporally distorting the initial representative. In practice, we use a parameter called *length fluctuation* defined as the maximum variability of the occurrence's length to the average length. For example, a ratio of 0.1 means that we resample the occurrences of the motif so they have lengths varying up to ±10% from the average length. The occurrences of all motifs are then randomly concatenated and spaced according to sparsity parameters. Finally, white Gaussian noise of standard deviation σ, and a random walk (to model local linear trends) are added to the signal.

We use baseline settings for all synthetic experiments, which can be found on GitHub. For each experiment, we vary one or more of these parameters according to the question we wish to answer (this setting is different for RQ2 since we vary the exact length of the time series, which is constant in other RQs).

3.3 Representative Motif Discovery Methods

The following motivations drive our selection of Motif Discovery methods: (i) Our collection of methods should have at least one representative from each of the main families of methods we presented earlier. (ii) Priority is given to methods that have represented a great advancement in the field [3, 88]. (iii) Our collection should contain recent approaches tackling a large panel of challenges enumerated in Section 2 [22, 69, 80]. (iv) We finally favor algorithms with available implementations or detailed code descriptions. These criteria led us to choose the following methods (summarized in Table 2):

Table 2: Our proposed collection of methods. K represents the number of motifs, w the window (or subsequence) length, R the range, r the range ratio, and n the length of the time series. *These methods use pruning strategies that reduce calculation time in most cases.

Methods	Parameters	Complexity (Worst Case)
SetFinder [3]	K, w, R	$O(n^3)$
LatentMotif [24]	K, w, R	$O(wn)$
STOMP [84]	K, w, r	$O(n^2)$
VALMOD [37]	$K, w_{\min}, w_{\max}, r$	$O((w_{\max} - w_{\min})n^2)^*$
PanMP [40]	$K, w_{\min}, w_{\max}, r$	$O((w_{\max} - w_{\min})n^2)$
k-Motiflets [67]	$k_{\max}, w_{\min}, w_{\max}$	$O(k_{\max}n^2 + nk_{\max}^2)$
PEPA [22]	$w_{\min}, K$	$O(Kn^2)$
A-PEPA [22]	$w_{\min}$	$O(Kn^2)$
GrammarViz [69]	K, w	$O(wn^2)$
MDL-Clust [61]	$w_{\min}, w_{\max}$	$O(n^3/w_{min} + (w_{max} - w_{min})n^2)^*$
LoCoMotif [80]	$K, w_{\min}, w_{\max}$	$O(n^2 \frac{w_{\max} - w_{\min}}{w_{\min}})$

SetFinder [3] finds the K-motif sets (c.f., Section 2) directly, based on a counting and separating principle. In practice, each subsequence is compared to every other, and the non-overlapping matches are counted. Then, each subsequence with a non-zero count is checked to ensure that its distance to another subsequence with a larger number of matches is greater than a given threshold.
LatentMotif [24] addresses a variant of the *K-Motifs* problem as a constrained optimization task, where the center of the motif is learned (the center does not need to be a subsequence of S but can belong to any element in $\mathbb{R}^n$). The objective and constraint functions are regularized to enable gradient ascent. The learned subsequences are then returned as the centers of the motif sets. A scan of the time series is conducted to identify all occurrences of each motif set. Non-overlapping subsequences within a distance R of the learned center are considered occurrences of the motif set.
STOMP [84] is a similarity-based method and proposes a fast computation of the Matrix Profile by efficiently leveraging the Fast Fourier Transform (FFT). Once the Matrix Profile is computed, the center of the Motif Set is defined as the subsequence with the minimum distance to another non-overlapping subsequence. A scan of the time-series subsequences is performed, and non-overlapping subsequences that are at a distance of less than R from the center are identified as occurrences of the corresponding Motif set [50].
PanMP [40] aims to generalize the Matrix Profile approach to detect patterns at varying time scales without requiring prior knowledge of the Motif size. To achieve this, the PanMatrixProfile—a matrix that contains Matrix Profiles for a range of window lengths—is computed. Based on distance and regardless of window size, the best non-overlapping Motif Pairs are then iteratively selected. The Motif sets are constructed from these selected Motif Pairs in the same way as in STOMP. Note that if the range of window sizes is restricted to a single value, PanMP is identical to STOMP.
VALMOD [37] has a similar goal to PanMP but employs a slightly different approach. It leverages pruning techniques to compute the Matrix Profile over a range of window lengths, ℓ. Motif Pairs are iteratively selected based on distance normalized by the square root of the window length. Motif sets are then built from these top Motif Pairs by identifying non-overlapping subsequences within a distance $< R$ from one of the two centers (see Section 2.3).
k-Motiflets [67] aims to discover motifs without needing to set a radius parameter R, unlike most other algorithms in our benchmark. Instead, the user specifies the desired number of occurrences k for the target motif. The method identifies the set of k non-overlapping subsequences with minimal extent, where extent is the maximum pairwise distance among elements in the set (see Section 2.3).
PEPA [22] extracts the motifs through three computational steps: (i) the time series is transformed into a graph with nodes representing subsequences and edges weighted by the distance between subsequences; (ii) persistent homology is applied to detect significant clusters of nodes, isolating them from nodes that correspond to irrelevant parts of the time series; and (iii) a post-processing step merges temporally adjacent subsequences within each cluster to form variable length motif sets. PEPA utilizes the LT-Normalized Euclidean distance [21], a distance invariant to linear trends.
A-PEPA [22] is variant of PEPA that does not require the user to define the exact number of motif sets and estimates it automatically.
Grammarviz [69] uses grammar induction methods for motif detection. In practice, the time series is discretized using SAX, and grammar induction techniques, such as *Sequitur* or *RE-PAIR*, are applied to the discretized series to identify grammar rules. The most frequent and representative grammar rules are selected, and occurrences of the various motifs are then extracted.
MDL-Clust [61] claims to perform clustering of subsequences. However, since clustering time series subsequences is generally ineffective [31], the authors propose disregarding data that does not fit into any cluster and avoiding overlapping subsequences. Thus, the output of MDL-CLust can be fully interpreted as motif sets. The method utilizes the MDL principle to form clusters. In each iteration, we can either create a new cluster (by selecting the first two members using a classic PairMotif algorithm), add a subsequence to an existing cluster, or merge two clusters. We select the operation that most effectively reduces the description length. The algorithm terminates when no usable data remains or further reduction in the time series description length is no longer possible.
LoCoMotif [80] addresses the challenge of variable length by searching for time-warped motifs at different time scales in the time series. In the first step, the *Self-Similarity Matrix* of the time series is utilized to construct paths based on a principle similar to Dynamic Time Warping (DTW). The paths with the highest accumulated similarity in this matrix are identified. In the second step, these subpaths are grouped to create candidate Motifs. The method then assesses the encoding capacity of these candidates using a quality score that combines the similarity between occurrences with the overall coverage of the Motif set.

3.4 Parameters Settings

All these methods share several common parameters (detailed in Table 2). These parameters are complex to set in practice and can strongly impact performances. In order to perform a fair comparison between all methods in our benchmarks, we set the values of these parameters to their optimal values (based on the exact characteristics of the time series in our benchmark). Overall, the parameters are the following:

The first is **the number of patterns** K to retrieve (SetFinder, Grammarviz, LatentMotifs, LoCoMotif, STOMP, VALMOD). In our experimental evaluation, we set this parameter to the exact number of patterns in the time series. One method in our benchmark only requires the **maximum number of occurrences ($k_{\max}$)** (Motiflets). In practice, we set this parameter to the maximum number of occurrences of all Motifs.

The second is the **radius** R (SetFinder, LatentMotifs). We can set this parameter for synthetic data, on which we can control the

radius. However, this parameter has to be estimated for real time series. In practice, for all the occurrences of a motif (the set of occurrences is noted M), we compute $R_k = \max_{S_i,S_j \in M} \text{dist}(S_i, S_j)/2$. The latter is the minimum radius for capturing all occurrences of M. We then compute the average radius for all motifs as follows: $\overline{R} = \frac{1}{K} \sum_{i=1}^{K} R_k$. We finally use $\overline{R}$ as the radius parameter.

The third parameter is the **radius ratio** r (STOMP, VALMOD, PanMP). This parameter is required for methods based on the best Motifs Pair to build the Motif Set. These methods propose a heuristic for the radius R, which is a ratio of the distance D between the two occurrences of the Motif Pair. The heuristic is the following: $R = r \times D$. In our benchmark, We take $r = 3$ as the default value.

The last parameter is the **window length (w)** (SetFinder, Grammarviz, LatentMotifs, MatrixProfile). In our benchmark, we set w as the average length of occurrences of all motifs. Some methods in our benchmark require a range between a **minimum length ($w_{\min}$)** and **maximum length ($w_{\max}$)** (VALMOD, LoCoMotifs, MDL, PEPA, Motiflets). In our experiments, we set these parameters to the minimum and maximum length of occurrences of all motifs.

3.5 Evaluation Measures

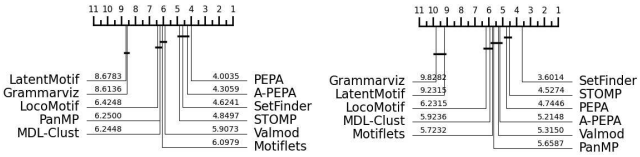
It is important to note that there is no consensus on how to empirically evaluate Motif Discovery in time series. As scalability is one of the most significant aspects of Motif Discovery, many research studies evaluated methods on the computation efficiency [12, 16, 34, 39, 49, 77, 82]. In terms of accuracy, the evaluation performed in the literature is mainly qualitative and the ability of algorithms to detect relevant patterns is visually assessed [12, 49, 77, 82]. Quantitatively, the accuracy is either assessed using the average distance between occurrences of a detected pattern [38, 62] or on the cardinality of the Motifs Sets [24, 67].

In this paper, we propose to use the metrics based on classic metrics for event detection tasks in time series [22, 76]. More precisely, we evaluate performance with range based-precision, recall, and f1-score metrics [76]. Nevertheless, the computation of these metrics requires pairing real and predicted motif sets. The optimal pairings maximize the total overlapping between real and predicted motif sets. The latter can be computed with the Hungarian matching algorithm [32, 66]. Then, the precision, recall, and f1-score computation rely on the optimal pairings and a threshold $\tau \in [0, 1]$ that controls the overlapping ratio. Any metric's score is the average of the individual metric score between paired motif sets; the averaging can be macro or weighted. For precision (resp. recall), a motif occurrence is counted as a true positive if the ratio between the overlap length and the predicted (resp. real) occurrence length is greater than the threshold τ. As in [22], this threshold is set to 50% for all experiments. Results for all threshold percentages, including plots for 25% and 75%, are however available on our repository. For clarity, we use the f1-score for comparisons throughout the paper, as it accounts for both precision and recall.

4 EXPERIMENTAL EVALUATION

We now describe our experimental analysis in detail. We use the benchmark described in Section 3 to answer 6 research questions arising directly from the challenges in Section 2.5. Our material is publicly available online (c.f. Artifact Availability).

The evaluation was performed on a server with Intel(R) Xeon(R) Gold 5220R CPU @ 2.20GHz, and 250 GB of RAM. When available,



(a) With REFIT and SIGN **(b) Without REFIT and SIGN**

Figure 5: Critical difference diagrams

we have used the original implementations of the algorithms, as is the case for LoCoMotif, Motiflets PEPA, A-PEPA, and Grammarviz. Otherwise, we have reimplemented the corresponding methods (STOMP, PANMP, VALMOD, MDL, LatentMotifs, SetFinder) in Python. All the methods are implemented in Python except Grammarviz, for which the authors' proposal was in JAVA. For the rest of this paper, we set a time-out threshold to 20 000 seconds. The performances of each method are averaged on the whole dataset for real data (RQ1), over 5 runs (i.e. 5 different generated time series) for execution time estimates (RQ2), for which we mainly want to have an order of magnitude, and on 100 runs for fscores estimates (RQ3 to RQ6), for which we want more precision.

Note: In the absence of length variation (as in RQ2 to 4, and 6), STOMP and PANMP produce identical and are labeled as STOMP/PANMP.

RQ1: Performances on real data

Are there any methods that stand out from the rest in terms of performance on real data datasets?

We start our experimental evaluation by measuring the performances (both in terms of f1-score and execution time) of the selected Motif Discovery methods on our collection of real labeled time series. All these time series contain motifs corresponding to actual temporal events and are relevant to the practical challenges enumerated in Section 2. Our evaluation is summarized in Table 3 (the empty cells correspond to methods that crashed or reached our time-out defined in the previous section). We also present critical difference diagrams, with (Figure 5(a)) and without (Figure 5(b)) REFIT and SIGN, showing the average rank of each method over the entire dataset. The dark lines represent cliques of methods with broadly similar performance, found using pairwise Wilcoxon tests.

Methods results vary greatly from one dataset to another. Except for Grammarviz, all methods perform well on at least one dataset, while Grammarviz stands out with significantly lower execution times. More precisely, we observe that the performances of all methods are low for REFIT and SIGN, which could be explained by a very low ratio of motifs per time series compared to other datasets (c.f. Table 1). The difficulties encountered by methods on these datasets opens an interesting research direction. On the other hand, the critical difference diagrams show that isolating a single best method is difficult. However, PEPA, A-PEPA, SetFinder and STOMP seem to stand out from the crowd on both diagrams. Finally, since the time series used in the experiments have very different characteristics, it remains challenging to determine which specific one has the most significant impact on each method's performance.

RQ1 Conclusion: *PEPA, A-PEPA, STOMP and SetFinder seem to have slightly better results on real data, according to critical diference diagrams. However, the variations in methods performances between the dataset show the importance of asking precise questions about*

Table 3: Fscore, and execution time in seconds of the methods on the real datasets. Standard deviations between parantheses. Empty cells corresponds to time-out. A version of the table with precision and recall is available on our repository.

dataset	metric	STOMP	PanMP	LoCoMotif	LatentMotif	MDL-Clust	k-Motiflets	PEPA	VALMOD	SetFinder	A-PEPA	GrammarViz
arm-coda	fscore	0.25 (0.15)	0.22 (0.10)	0.17 (0.17)	0.27 (0.14)	**0.66** (0.25)	0.03 (0.07)	0.29 (0.14)	0.29 (0.15)	0.20 (0.05)	0.29 (0.17)	0.01 (0.02)
	Exec. time	0.5 (0.06)	170 (63)	18 (8)	30 (9)	555 (159)	154 (42)	2 (0.3)	303 (80)	1.5 (0.5)	2 (0.3)	**0.3** (0.00)
mitdb	fscore	0.50 (0.20)	0.14 (0.22)	0.12 (0.18)	0.29 (0.24)	0.33 (0.15)	0.40 (0.37)	0.41 (0.30)	0.17 (0.23)	**0.55** (0.17)	0.51 (0.19)	0.00 (0.00)
	Exec. time	2.9 (0.01)	934 (600)	1252 (3837)	14 (8)	4178 (1483)	16396 (10413)	11 (0.4)	1762 (1273)	14 (2.3)	11 (0.4)	**0.41** (0.02)
mitdb1	fscore	0.63 (0.19)	0.69 (0.26)	0.29 (0.14)	0.14 (0.14)	0.18 (0.07)	0.44 (0.37)	0.46 (0.34)	0.66 (0.25)	**0.77** (0.10)	0.36 (0.20)	0.00 (0.00)
	Exec. time	3 (0.05)	187 (105)	76 (8)	7 (1.5)	1133 (254)	3157 (1918)	11(0.5)	156(48)	12 (1.2)	10 (0.5)	**0.42** (0.02)
ptt-ppg	fscore	0.49 (0.18)	0.53 (0.23)	0.38 (0.16)	0.27 (0.17)	0.18 (0.07)	0.61 (0.26)	0.68 (0.12)	0.54 (0.23)	**0.69** (0.05)	0.43 (0.16)	0.00 (0.01)
	Exec. time	3 (0.6)	270 (200)	102 (17)	8 (2.8)	1261 (279)	4598 (2630)	11 (0.2)	204 (86)	23 (3)	12 (1.4)	**0.4** (0.02)
JIGSAWMaster	fscore	0.26 (0.10)	0.10 (0.12)	**0.33** (0.10)	0.26 (0.12)	0.23 (0.08)	0.13 (0.08)	0.18 (0.09)	0.17 (0.09)	0.23 (0.04)	0.20 (0.09)	0.10 (0.05)
	Exec. time	0.9 (0.8)	420 (520)	318 (665)	7 (6)	2214 (2147)	660 (669)	4 (3)	1208 (1038)	5 (5)	4 (3)	**0.31** (0.04)
JIGSAWSlave	fscore	0.25 (0.12)	0.05 (0.07)	**0.33** (0.12)	0.24 (0.10)	0.23 (0.06)	0.15 (0.10)	0.17 (0.08)	0.20 (0.10)	0.22 (0.05)	0.18 (0.08)	0.10 (0.06)
	Exec. time	0.87 (0.68)	343 (300)	189 (267)	6 (4)	2005 (1812)	590 (512)	4 (3)	1453 (1459)	4.7 (4)	4 (2)	**0.31** (0.03)
REFIT	fscore	0.00 (0.03)	-	-	0.03 (0.08)	-	-	0.14 (0.12)	-	-	**0.16** (0.15)	0.00 (0.00)
	Exec. time	500 (96)	-	-	230 (122)	-	-	1280(100)	-	-	1310 (120)	**63**(12)
SIGN	fscore	0.06 (0.04)	-	-	0.14 (0.09)	-	-	0.17 (0.03)	-	-	**0.20**(0.06)	0.10 (0.07)
	Exec. time	300 (25)	-	-	50(10)	-	-	900 (85)	-	-	900 (88)	**5** (18)

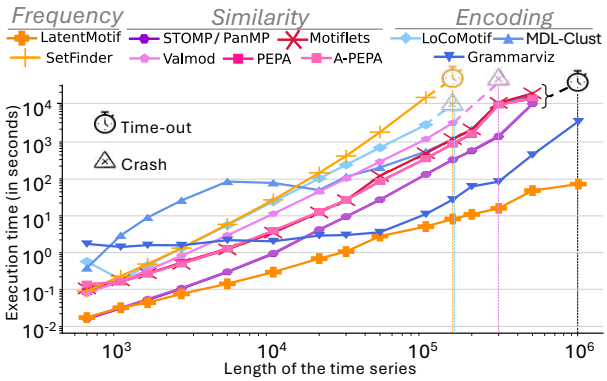


Figure 6: Execution time versus the time series length

which time series characteristics influence the performance of the algorithms. Thus, in the following sections, we benefit from our synthetic generator in identifying specific challenges. In practice, we generate time series with the default parameters detailed in Section 3, and we vary only the parameter of interest for a corresponding challenge, such as the number of different motifs or the noise amplitude.

RQ2: Scalability for long time-series

Are the methods capable of solving the problem in a reasonable amount of time for a relatively long time series ?

In this section, we measure the influence of the length of the time series on the execution time of the methods. In practice, we generate time series containing 5 occurrences of a single motif of length 100 and we vary the total length of the time series from 600 to 1 000 000 samples. Firstly, the execution times obtained are consistent with the theoretical complexities of the algorithms in Table 2. We see a quadratic growth for all methods except SetFinder (cubic growth) and LatentMotif (linear growth). Moreover, some methods scale much better than others. LoCoMotif crash for 150,000 samples and Valmod for 200,000. On the other hand, SetFinder reaches our time-out threshold of 200,000 samples, while all the other methods except LatentMotif and Grammarviz reach 1,000,000 samples.

RQ2 Conclusion: *Grammarviz and LatentMotif are the most scalable methods in terms of computational efficiency for handling long time series. STOMP, PanMP, MDL-Clust, PEPA, and A-PEPA have acceptable execution times up to 500,000 samples, while the other methods crash or exceed the timeout before that.*

RQ3: Presence of several different motifs

Are the methods robust to a high number of different motifs?

In this section, we evaluate the robustness of methods to a variation of the number of different motifs. In practice, we generate time series with the basic setting mentioned in Section 3.2, except for the number of different motifs, which varies from 1 to 50. Figure 7(a) depicts the average f1score on 100 runs for each method. For example, we show the synthetic time series generated for 1 motif (top-left), 5 motifs (top-middle), and 10 motifs (top-right).

Methods using a fixed radius (SetFinder, LatentMotif) see a performance drop as the number of motifs increases. In contrast, STOMP, PANMP, and VALMOD, which adjust the radius based on PairMotifs' distances, are less affected—VALMOD even remains stable due to its different motif definition, making it more suitable in the presence of many motifs. MDL-Clust and A-PEPA also decline in performance, because they estimate the number of motifs, and the more motifs there are, the more complicated this estimation becomes. Surprisingly, LoCoMotif performs poorly with few motifs but improves significantly as more motifs are added. This is due to its lack of subsequence normalization, initially misidentifying flat areas as motifs. As the number of motifs increases, true motifs are detected alongside flat areas, but the latter have a much smaller impact on the final score.

RQ3 conclusion: VALMOD, PEPA, and Grammarviz are particularly robust to the number of motifs. In addition, Locomotif has good performances for a large number of motifs. However, all of these algorithms require the number of motifs K as input.

RQ4: Motifs cardinality

Are the methods capable of finding all occurrences of the most relevant motifs independently of their cardinality? (RQ4.1)
Are the methods capable of detecting several motifs when there is a cardinality unbalance? (RQ4.2)

In this section, we first evaluate the ability of methods to detect the occurrence of motifs with different cardinalities (RQ4.1). In practice, we generate time series with the setting mentioned in Section 3.2, except for the number of motif occurrences, which

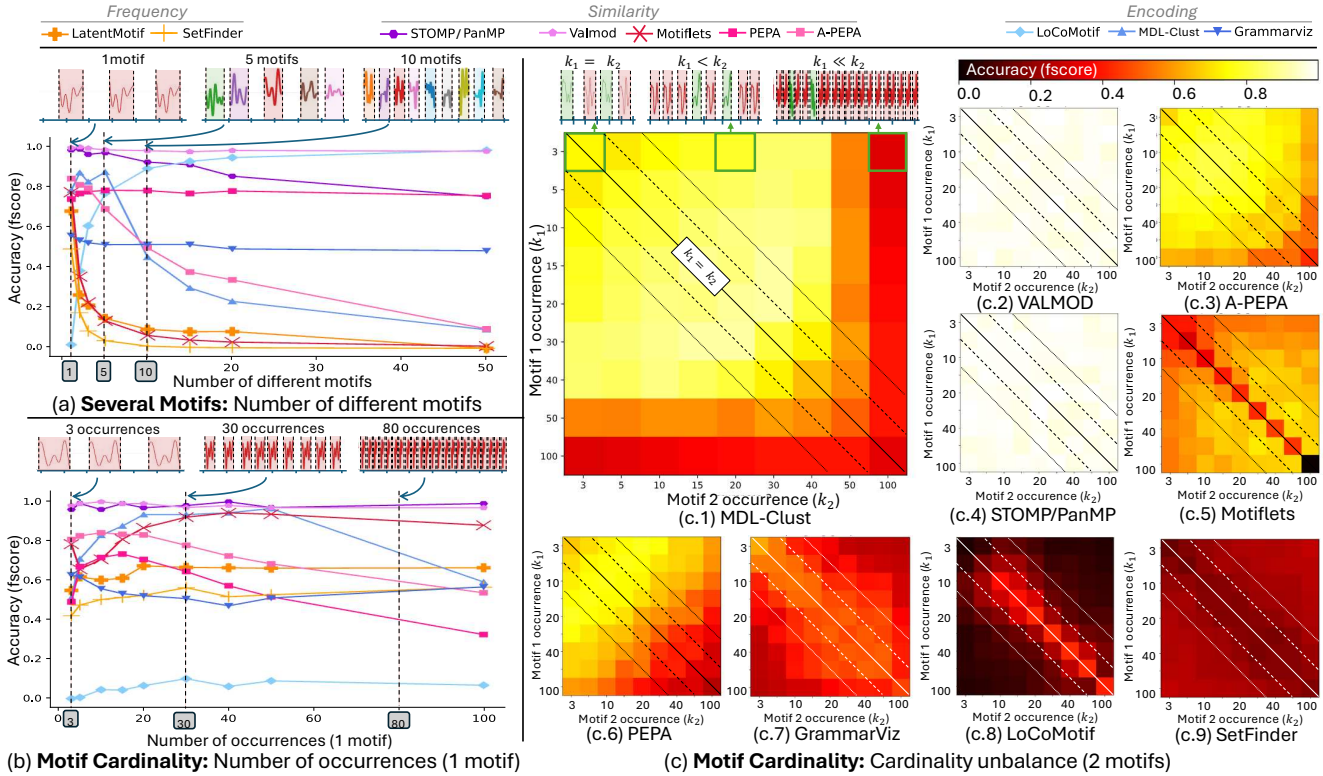


Figure 7: Influence of the number of motifs (top-left); the number of occurrences (bottom-left); the cardinality unbalance (right).

varies from 2 to 100. Figure 7(b) depicts the average f1score on 100 runs for each method. We show synthetic time series generated for 3 (top-left), 30 (top-middle), and 80 occurrences (top-right).

Most methods (STOMP, Valmod, SetFinder, LatentMotif, Grammarviz and LoCoMotif) show stable performance regardless of the number of occurrences. PEPA and A-PEPA perform best around 10 occurrences. These methods tend to overestimate motif lengths, but the impact on the score diminishes with more occurrences. However, as the number of occurrences increases, they are more likely to merge distinct, closely spaced occurrences, which lowers the score. MDL-Clust performs best around 50 occurrences, as its encoding-based approach benefits from more occurrences, making motifs more significant and easier to detect. However, with too many occurrences, the algorithm tends to create subclusters for the same motif, reducing performance. Motiflets performs well with both few and many occurrences, though it shows a slight dip in accuracy around 5 occurrences. This is because it uses an elbow technique to estimate the number of occurrences, which has difficulty in accurately estimating the exact number of occurrences. However, as the number of occurrences increases, missing a few has less impact on overall performance.

RQ4.1 conclusion: *VALMOD, PanMP, STOMP, Grammarviz, LatentMotif and SetFinder are particularly robust to the number of occurrences. Motiflets has good performances when motifs sets have many occurences.*

We now evaluate the impact of an unbalanced cardinality between different motifs, i.e., a difference in the number of occurrences between two different motifs (RQ4.2). In practice, we generate time series with the baseline settings, except that there are 2 motifs and that the respective number of occurrences of these 2 motifs varies between 3 and 100 to obtain all possible combinations. Figure 7(c) depicts the average f1score on 100 runs for several methods (the missing methods are provided in our repository). The diagonal corresponds to cases where the two motifs have the same number of occurrences. For example, we show synthetic time series generated with 3 and 3 (top left), 3 and 20 (top middle), and 3 and 100 (top right) occurrences of motif 1 and motif 2.

Figure 7(c) highlights four behaviors: (i) methods for which the variation in the occurrences of the two motifs has almost no impact (VALMOD, STOMP, and SetFinder) (ii) methods for which the impact is not related to the difference in occurrences between the two motifs but rather to the number of occurrences itself (PEPA, MDL-Clust, and A-PEPA). (iii) methods for which performance decreases when we move away from the diagonal -i.e. perfectly balanced cardinalities- (LoCoMotif and Grammarviz). (iv) Motiflets (Figure 7(c.6)) for which performances increase when we move away from the diagonal -i.e. unbalanced cardinalities. This last behavior stems from Motiflets' design, which aims to find the top 1 motif for a fixed number of occurrences. We remind that the exact number of occurences is determined by an heuristic. When multiple motifs with different occurrence counts are present, the heuristics may identify multiple optimal values, allowing the detection of each motif sets. This explains why Motiflets performs better outside the diagonal (i.e. with unbalanced cardinalities).

RQ4.2 conclusion: *VALMOD, STOMP, and PanMP are particularly robust to unbalanced cardinality of motif sets.*

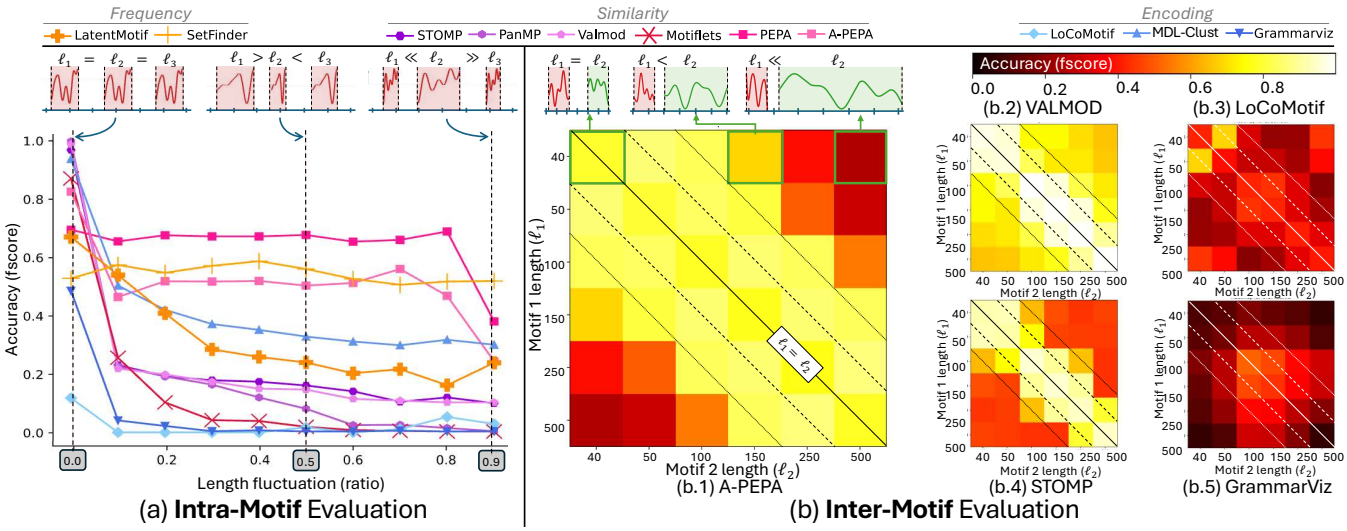


Figure 8: Influence of Intra-Motif variable length(left); influence of Inter-Motif variable length (right)

RQ5: Variable length

Are the methods robust to the presence of an Intra-Motif variable length due to temporal deformations of the initial motif ? (RQ5.1)
Are the methods able to detect motifs of different timescales? (RQ5.2)
In this section, we first evaluate the robustness of methods to the presence of an Intra-Motif variable length due to temporal deformations of the initial motif (RQ5.1). In practice, we generate time series with the baseline settings except for the *length fluctuation* parameter described in Section 3.2, which varies from 0 to 0.9. Figure 7(b) depicts each method's average f1score on 100 runs. As examples, we show synthetic time series generated for a length fluctuation ratio of 0.0 (top-left), 0.5 (top-middle), and 0.9 (top-right).

As expected, STOMP, PANMP, LatentMotif, and Motiflets experience a performance drop even with slight length variations, as they rely on rigid distances and enforce uniform occurrence lengths within the same motif set. Surprisingly, SetFinder, despite having the same constraint, maintains relatively stable performance. Among methods designed to handle intra-motif length variation, only PEPA and A-PEPA maintain good performance even with significant fluctuations, while Grammarviz and LoCoMotif experience a drop even with slight variations. PEPA and A-PEPA address this issue by merging overlapping subsequences into a single occurrence, which appears to be the most effective approach.

RQ5.1 conclusion: *PEPA is the most robust method to length variation, followed by A-PEPA and SetFinder. The latter maintains a constant performance, although not as high as the two others.*

We now evaluate the methods' ability to detect motifs present at different time scales, meaning that motifs with different average lengths represent different time scales (e.g. one second or one minute) (RQ5.2). In practice, we generate time series according to the baseline settings, except that there are 2 motifs and the average lengths of motif occurrences vary. We fix the average length of the first motif between the following values: 40,50,100,150,250,500. The average length of the second motif varies within the same range of values to obtain all possible combinations. Figure 8(b) depicts the average f1score on 100 runs for each method and a combination of average lengths. The diagonal corresponds to cases in which the two motifs have the same length. As examples, we show synthetic time series generated with motif 1 and motif 2 of length 40 (top-left), motif 1 of length 40 and motif 2 of length 150 (top-middle), motif 1 of length 40, and motif 2 of length 500 (top-right). We only show the results for 5 methods (the results of the remaining methods can be found in our repository).

Figure 8(b) highlights three behaviors: (i) methods allowing some flexibility in occurrence lengths across motif sets, which maintain good performance outside the diagonal (i.e. with Inter-Motifs length variation) but may drop at the extremes (VALMOD in Fig 8(b.2), A-PEPA in Fig 8(b.1)); (ii) methods without this flexibility, performing well on the diagonal (i.e. without Inter-Motifs length variation) but experiencing a sharp drop elsewhere (STOMP in Fig 8(b.4)); and (iii) methods with consistently low performance and no clear performance pattern (LoCoMotif in Fig 8(b.3), Grammarviz in Fig 8(b.5)).

RQ5.2 conclusion: *MDL-Clust, STOMP and Motiflets are the most suitable methods for small timescale variations between motifs. Only PEPA, A-PEPA and VALMOD maintain relatively good performance as soon as the scale variation increase. Overall, only VALMOD maintains good performance for extreme cases.*

RQ6: Spatial deformations

Are the methods robust to linear trends in the time series ? (RQ6.1)
Are the methods robust to noise in the time series ? (RQ6.2)

In this section, we first evaluate the robustness of methods to the presence of linear trends of different amplitudes (RQ6.1). In practice, we generate time series with the baseline settings and add a linear trend modeled by a background random walk. This random walk is generated as the cumulative sum of a Gaussian noise of standard deviation varying from 0 to 50. Figure 9(b) depicts each method's average f1score on 100 runs. For example, we show synthetic time series generated for a walk amplitude level of 0 (top-left), 5 (top-middle), and 50 (top-right).

STOMP, PANMP, VALMOD, and LatentMotif experience a performance decline, varying in speed, when walk amplitude is introduced—an expected outcome given their reliance on distance

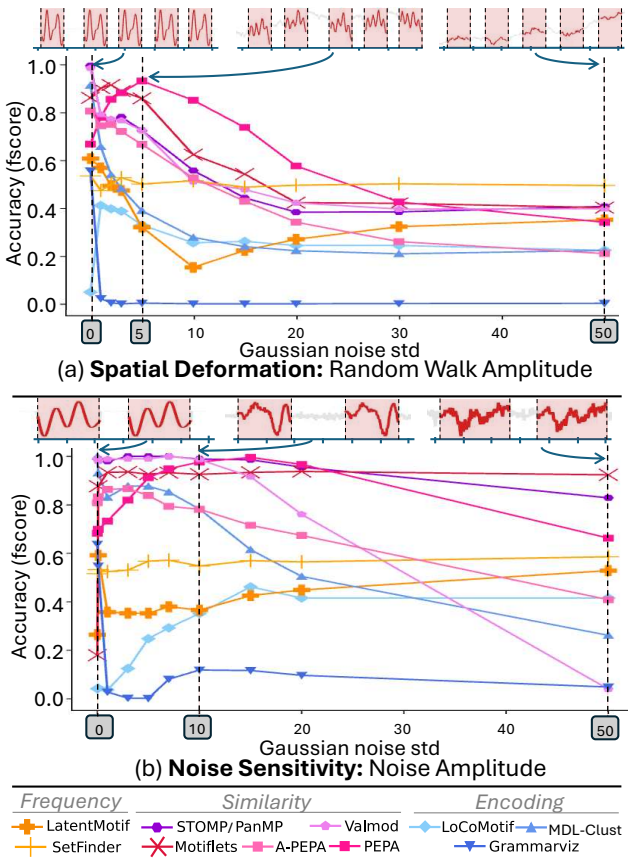


Figure 9: Influence of local linear trends (top); and noise (bottom)

measures highly sensitive to linear trends. MDL-Clust and Grammarviz show a predictable drop due to their use of discretization, which quickly degrades. Grammarviz suffers a sharp decline, while MDL-Clust's drop is more gradual, likely due to differences in their discretization methods. In contrast, Motiflets and LoCoMotif see a performance increase with small random walk before declining. This occurs because mild random walk helps distinguish motif-free zones, correcting previous misinterpretations. However, for high random walk amplitude, motifs are harder to recognize, leading to a performance drop. PEPA follows a similar pattern, with an initial improvement lasting longer than for other algorithms. This is due to its use of the LT-Normalized Euclidean distance, which provides some invariance to local linear trends, allowing it to correctly detect motifs even at higher walk amplitudes.

RQ6.1 conclusion: *SetFinder shows consistent performances in the presence of random walk. Motiflets, PEPA, and LoCoMotif maintain correct performance even for high random walk amplitudes.*

We now evaluate the robustness of methods to noise. For that purpose, we generate time series according to the baseline settings except for the noise amplitude. A Gaussian noise of standard deviation varying from 0.01 to 50 is added to the time series. This range corresponds to signal-to-noise ratio (SNR) varying from 75 to 1,5.

SetFinder maintains stable performance across all noise levels, while LatentMotif, despite an initial drop, remains fairly stable and even improves as noise increases. In contrast, Grammarviz and

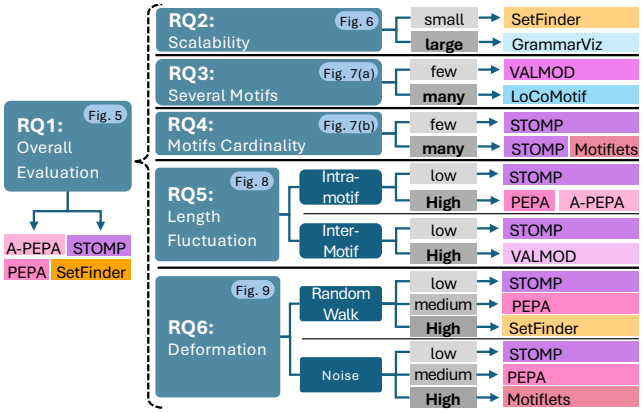


Figure 10: Overall guidelines inferred from our evaluation

MDL-Clust both experience performance drops due to the sensitivity of discretizations to noise; however, Grammarviz declines sharply, whereas MDL-Clust shows a more gradual decrease. Similarly, STOMP and VALMOD also see performance declines, but STOMP's stricter occurence selection criteria enable it to perform better than VALMOD. On the other hand, LoCoMotif and Motiflets benefit from higher noise levels, as the noise helps distinguish non-motif zones, thereby preventing misclassification. Notably, Motiflets show significant improvement in these conditions. Finally, PEPA and A-PEPA initially improve with slight noise but then decline, with A-PEPA dropping more quickly due to its tendency to overestimate the number of motifs sets as noise increases.

RQ6.2 conclusion: *STOMP, PANMP, and Motiflets are the most suitable for very noisy time series. If the noise is non-zero without being excessively high, one can also consider using VALMOD or PEPA.*

5 CONCLUSIONS

Motif Discovery is a challenging task with significant applications across various fields. Given the numerous methods available in the literature, proper evaluation is crucial. In this paper, we conducted a literature review and a comprehensive evaluation under challenging scenarios. While some methods (e.g., PEPA, A-PEPA, STOMP, SETFINDER) show better overall performance on real data, no single method can address all challenges effectively. Performance largely depends on the specific characteristics of the time series. To move beyond this broad conclusion, we designed experiments to isolate and measure the impact of individual time series characteristics. Our findings address six key research questions and provide guidelines to help users select the most suitable method for their needs (summarized in Figure 10). However, our experiments focused on simple scenarios to deliver clear insights. We considered time series with complex combinations of non-trivial characteristics in the context of RQ1 on real-world data. In this specific case, we cannot isolate the influence of individual characteristics. Further research could investigate scenarios combining multiple challenges, opening avenues for future studies. We believe that Motif Discovery remains an open problem, as no single method is uniformly better. Even on some of our arguably simple datasets, there is still room for progress. We hope this work can provide valuable insights and contribute to ongoing efforts to improve Motif Discovery.

REFERENCES

[1] Rakesh Agrawal, Christos Faloutsos, and Arun Swami. 1993. Efficient similarity search in sequence databases. In *Foundations of Data Organization and Algorithms: 4th International Conference, FODO'93 Chicago, Illinois, USA, October 13–15, 1993 Proceedings 4*. Springer, 69–84.

[2] Sara Alaee, Kaveh Kamgar, and Eamonn Keogh. 2020. Matrix profile XXII: exact discovery of time series motifs under DTW. In *2020 IEEE international conference on data mining (ICDM)*. IEEE, 900–905.

[3] Anthony Bagnall, Jon Hills, and Jason Lines. 2014. Finding motif sets in time series. *arXiv preprint arXiv:1407.3685* (2014).

[4] Anthony Bagnall, Jason Lines, Aaron Bostrom, James Large, and Eamonn Keogh. 2017. The great time series classification bake off: a review and experimental evaluation of recent algorithmic advances. *Data Min. Knowl. Discov.* 31, 3 (May 2017), 606–660. https://doi.org/10.1007/s10618-016-0483-9

[5] Kevin Bascol, Rémi Emonet, Elisa Fromont, and Jean-Marc Odobez. 2016. Unsupervised interpretable pattern discovery in time series using autoencoders. In *Structural, Syntactic, and Statistical Pattern Recognition: Joint IAPR International Workshop, S+ SSPR 2016, Mérida, Mexico, November 29-December 2, 2016, Proceedings*. Springer, 427–438.

[6] Paul Boniol, Mohammed Meftah, Emmanuel Remy, Bruno Didier, and Themis Palpanas. 2023. dCNN/dCAM: anomaly precursors discovery in multivariate time series with deep convolutional neural networks. *Data-Centric Engineering* 4 (2023), e30. https://doi.org/10.1017/dce.2023.25

[7] Jeremy Buhler and Martin Tompa. 2001. Finding motifs using random projections. In *Proceedings of the fifth annual international conference on Computational biology*. 69–76.

[8] Nuno Castro and Paulo Azevedo. 2010. Multiresolution motif discovery in time series. In *Proceedings of the 2010 SIAM international conference on data mining*. SIAM, 665–676.

[9] Joe Catalano, Tom Armstrong, and Tim Oates. 2006. Discovering patterns in real-valued time series. In *European Conference on Principles of Data Mining and Knowledge Discovery*. Springer, 462–469.

[10] Matteo Ceccarello and Johann Gamper. 2022. Fast and Scalable Mining of Time Series Motifs with Probabilistic Guarantees. *Proceedings of the VLDB Endowment* 15, 13 (2022), 3841–3853.

[11] Kin-Pong Chan and Ada Wai-Chee Fu. 1999. Efficient time series matching by wavelets. In *Proceedings 15th International Conference on Data Engineering (Cat. No. 99CB36337)*. IEEE, 126–133.

[12] Bill Chiu, Eamonn Keogh, and Stefano Lonardi. 2003. Probabilistic discovery of time series motifs. In *Proceedings of the ninth ACM SIGKDD international conference on Knowledge discovery and data mining*. 493–498.

[13] Sylvain W. Combettes, Paul Boniol, Antoine Mazarguil, Danping Wang, Diego Vaquero-Ramos, Marion Chauveau, Laurent Oudre, Nicolas Vayatis, Pierre-Paul Vidal, Alexandra Roren, and Marie-Martine Lefèvre-Colau. 2024. Arm-CODA: A Data Set of Upper-limb Human Movement During Routine Examination. *Image Processing On Line* 14 (2024), 1–13.

[14] Sahar Deppe and Volker Lohweg. 2015. Shift-Invariant Feature Extraction for Time-Series Motif Discovery. In *25. Workshop Computational Intelligence VDI/VDE-Gesellschaft Mess-und Automatisierungstechnik (GMA)*.

[15] Richard Durbin, Sean R Eddy, Anders Krogh, and Graeme Mitchison. 1998. *Biological sequence analysis: probabilistic models of proteins and nucleic acids*. Cambridge university press.

[16] Pedro G Ferreira, Paulo J Azevedo, Cândida G Silva, and Rui MM Brito. 2006. Mining approximate motifs in time series. In *Discovery Science: 9th International Conference, DS 2006, Barcelona, Spain, October 7-10, 2006. Proceedings 9*. Springer, 89–101.

[17] Yifeng Gao and Jessica Lin. 2018. Exploring variable-length time series motifs in one hundred million length scale. *Data Mining and Knowledge Discovery* 32 (2018), 1200–1228.

[18] Yifeng Gao and Jessica Lin. 2019. HIME: discovering variable-length motifs in large-scale time series. *Knowledge and Information Systems* 61 (2019), 513–542.

[19] Yixin Gao, S. Vedula, Carol E. Reiley, N. Ahmidi, B. Varadarajan, Henry C. Lin, L. Tao, L. Zappella, B. Béjar, D. Yuh, C. C. Chen, R. Vidal, S. Khudanpur, and Gregory Hager. 2014. JHU-ISI Gesture and Skill Assessment Working Set (JIGSAWS) : A Surgical Activity Dataset for Human Motion Modeling.

[20] Xianping Ge and Padhraic Smyth. 2000. Deformable Markov model templates for time-series pattern matching. In *Proceedings of the sixth ACM SIGKDD international conference on Knowledge discovery and data mining*. 81–90.

[21] Thibaut Germain, Charles Truong, and Laurent Oudre. 2024. Linear-trend normalization for multivariate subsequence similarity search. In *2024 IEEE 40th International Conference on Data Engineering Workshops (ICDEW)*. IEEE, 167–175.

[22] Thibaut Germain, Charles Truong, and Laurent Oudre. 2024. Persistence-based motif discovery in time series. *IEEE Transactions on Knowledge and Data Engineering* (2024).

[23] Ary L Goldberger, Luis AN Amaral, Leon Glass, Jeffrey M Hausdorff, Plamen Ch Ivanov, Roger G Mark, Joseph E Mietus, George B Moody, Chung-Kang Peng, and H Eugene Stanley. 2000. PhysioBank, PhysioToolkit, and PhysioNet: components of a new research resource for complex physiologic signals. *circulation* 101, 23 (2000), e215–e220.

[24] Josif Grabocka, Nicolas Schilling, and Lars Schmidt-Thieme. 2016. Latent time-series motifs. *ACM Transactions on Knowledge Discovery from Data (TKDD)* 11, 1 (2016), 1–20.

[25] Gerald Z Hertz and Gary D. Stormo. 1999. Identifying DNA and protein patterns with statistically significant alignments of multiple sequences. *Bioinformatics (Oxford, England)* 15, 7 (1999), 563–577.

[26] Kyle L Jensen, Mark P Styczynski, Isidore Rigoutsos, and Gregory N Stephanopoulos. 2006. A generic motif discovery algorithm for sequential data. *Bioinformatics* 22, 1 (2006), 21–28.

[27] Mohammed Kadous. 1995. Australian Sign Language Signs. UCI Machine Learning Repository. DOI: https://doi.org/10.24432/C5XG6C.

[28] Konstantinos Kalpakis, Dhiral Gada, and Vasundhara Puttagunta. 2001. Distance measures for effective clustering of ARIMA time-series. In *Proceedings 2001 IEEE international conference on data mining*. IEEE, 273–280.

[29] Eamonn Keogh, Kaushik Chakrabarti, Michael Pazzani, and Sharad Mehrotra. 2001. Dimensionality reduction for fast similarity search in large time series databases. *Knowledge and information Systems* 3 (2001), 263–286.

[30] Eamonn Keogh, Kaushik Chakrabarti, Michael Pazzani, and Sharad Mehrotra. 2001. Locally adaptive dimensionality reduction for indexing large time series databases. In *Proceedings of the 2001 ACM SIGMOD international conference on Management of data*. 151–162.

[31] Eamonn Keogh and Jessica Lin. 2005. Clustering of time-series subsequences is meaningless: implications for previous and future research. *Knowledge and information systems* 8 (2005), 154–177.

[32] Harold W Kuhn. 1955. The Hungarian method for the assignment problem. *Naval research logistics quarterly* 2, 1-2 (1955), 83–97.

[33] Hoang Thanh Lam, Ninh Dang Pham, and Toon Calders. 2011. Online discovery of top-k similar motifs in time series data. In *Proceedings of the 2011 SIAM International Conference on Data Mining*. SIAM, 1004–1015.

[34] Yuhong Li, Hou U Leong, Man Lung Yiu, and Zhiguo Gong. 2015. Quick-motif: An efficient and scalable framework for exact motif discovery. In *2015 IEEE 31st International Conference on Data Engineering*. IEEE, 579–590.

[35] Yuan Li and Jessica Lin. 2010. Approximate variable-length time series motif discovery using grammar inference. In *Proceedings of the Tenth International Workshop on Multimedia Data Mining*. 1–9.

[36] Yuan Li, Jessica Lin, and Tim Oates. 2012. Visualizing variable-length time series motifs. In *Proceedings of the 2012 SIAM international conference on data mining*. SIAM, 895–906.

[37] Michele Linardi, Yan Zhu, Themis Palpanas, and Eamonn Keogh. 2018. Matrix profile X: VALMOD-scalable discovery of variable-length motifs in data series. In *Proceedings of the 2018 international conference on management of data*. 1053–1066.

[38] Zheng Liu, Jeffrey Xu Yu, Xuemin Lin, Hongjun Lu, and Wei Wang. 2005. Locating motifs in time-series data. In *Pacific-Asia Conference on Knowledge Discovery and Data Mining*. Springer, 343–353.

[39] JLEKS Lonardi and Pranav Patel. 2002. Finding motifs in time series. In *Proc. of the 2nd Workshop on Temporal Data Mining*. 53–68.

[40] Frank Madrid, Shima Imani, Ryan Mercer, Zachary Zimmerman, Nader Shakibay, and Eamonn Keogh. 2019. Matrix profile xx: Finding and visualizing time series motifs of all lengths using the matrix profile. In *2019 IEEE International conference on big knowledge (ICBK)*. IEEE, 175–182.

[41] Philip Mehrgardt, Matloob Khushi, Simon Poon, and Anusha Withana. 2022. Pulse Transit Time PPG Dataset. *PhysioNet* 10 (2022), e215–e220.

[42] David Minnen, Charles L Isbell, Irfan Essa, and Thad Starner. 2007. Discovering multivariate motifs using subsequence density estimation and greedy mixture learning. In *Proceedings of the national conference on artificial intelligence*, Vol. 22. Menlo Park, CA; Cambridge, MA; London; AAAI Press; MIT Press; 1999, 615.

[43] David Minnen, Thad Starner, Irfan Essa, and Charles Isbell. 2006. Discovering characteristic actions from on-body sensor data. In *2006 10th IEEE international symposium on wearable computers*. IEEE, 11–18.

[44] David Minnen, Thad Starner, Irfan A Essa, and Charles Lee Isbell Jr. 2007. Improving Activity Discovery with Automatic Neighborhood Estimation.. In *IJCAI*, Vol. 7. 2814–2819.

[45] Yasser Mohammad and Toyoaki Nishida. 2014. Exact discovery of length-range motifs. In *Asian Conference on Intelligent Information and Database Systems*. Springer, 23–32.

[46] Abdullah Mueen and Nikan Chavoshi. 2015. Enumeration of time series motifs of all lengths. *Knowledge and Information Systems* 45, 1 (2015), 105–132.

[47] Abdullah Mueen and Eamonn Keogh. 2010. Online discovery and maintenance of time series motifs. In *Proceedings of the 16th ACM SIGKDD international conference on Knowledge discovery and data mining*. 1089–1098.

[48] Abdullah Mueen, Eamonn Keogh, and Nima Bigdely-Shamlo. 2009. Finding time series motifs in disk-resident data. In *2009 Ninth IEEE International Conference on Data Mining*. IEEE, 367–376.

[49] Abdullah Mueen, Eamonn Keogh, Qiang Zhu, Sydney Cash, and Brandon Westover. 2009. Exact discovery of time series motifs. In *Proceedings of the 2009 SIAM international conference on data mining*. SIAM, 473–484.

[50] Abdullah Mueen, Sheng Zhing, Yan Zhu, Michael Yeh, Kaveh Kamgar, Krishnamurthy Viswanathan, Chetan Gupta, and Eamonn Keogh. 2022. The Fastest Similarity Search Algorithm for Time Series Subsequences under Euclidean Distance. http://www.cs.unm.edu/~mueen/FastestSimilaritySearch.html.

[51] David Murray, Lina Stankovic, and Vladimir Stankovic. 2017. An electrical load measurements dataset of United Kingdom households from a two-year longitudinal study. *Scientific data* 4, 1 (2017), 1–12.

[52] Fabian Kai-Dietrich Noering, Yannik Schroeder, Konstantin Jonas, and Frank Klawonn. 2021. Pattern discovery in time series using autoencoder in comparison to nonlearning approaches. *Integrated Computer-Aided Engineering* 28, 3 (2021), 237–256.

[53] Pawan Nunthanid, Vit Niennattrakul, and Chotirat Ann Ratanamahatana. 2011. Discovery of variable length time series motif. In *The 8th Electrical Engineering/Electronics, Computer, Telecommunications and Information Technology (ECTI) Association of Thailand-Conference 2011*. IEEE, 472–475.

[54] Pawan Nunthanid, Vit Niennattrakul, and Chotirat Ann Ratanamahatana. 2012. Parameter-free motif discovery for time series data. In *2012 9th International Conference on Electrical Engineering/Electronics, Computer, Telecommunications and Information Technology*. IEEE, 1–4.

[55] Tim Oates. 2002. PERUSE: An unsupervised algorithm for finding recurring patterns in time series. In *2002 IEEE International Conference on Data Mining, 2002. Proceedings.* IEEE, 330–337.

[56] Themis Palpanas. 2015. Data Series Management: The Road to Big Sequence Analytics. *SIGMOD Rec.* 44, 2 (Aug. 2015), 47–52. https://doi.org/10.1145/2814710.2814719

[57] John Paparrizos and Luis Gravano. 2016. k-Shape: Efficient and Accurate Clustering of Time Series. *SIGMOD Rec.* 45, 1 (June 2016), 69–76. https://doi.org/10.1145/2949741.2949758

[58] John Paparrizos, Yuhao Kang, Paul Boniol, Ruey S. Tsay, Themis Palpanas, and Michael J. Franklin. 2022. TSB-UAD: an end-to-end benchmark suite for univariate time-series anomaly detection. *Proc. VLDB Endow.* 15, 8 (April 2022), 1697–1711. https://doi.org/10.14778/3529337.3529354

[59] Adrien Petralia, Philippe Charpentier, Paul Boniol, and Themis Palpanas. 2023. Appliance Detection Using Very Low-Frequency Smart Meter Time Series. In *Proceedings of the 14th ACM International Conference on Future Energy Systems* (Orlando, FL, USA) *(e-Energy '23)*. Association for Computing Machinery, New York, NY, USA, 214–225. https://doi.org/10.1145/3575813.3595198

[60] Pavel A Pevzner, Sing-Hoi Sze, et al. 2000. Combinatorial approaches to finding subtle signals in DNA sequences.. In *ISMB*, Vol. 8. 269–278.

[61] Thanawin Rakthanmanon, Eamonn J Keogh, Stefano Lonardi, and Scott Evans. 2012. MDL-based time series clustering. *Knowledge and information systems* 33 (2012), 371–399.

[62] Simona Rombo and Giorgio Terracina. 2004. Discovering representative models in large time series databases. In *Flexible Query Answering Systems: 6th International Conference, FQAS 2004, Lyon, France, June 24-26, 2004. Proceedings 6*. Springer, 84–97.

[63] Chuitian Rong, Ziliang Chen, Chunbin Lin, and Jianming Wang. 2020. Motif Discovery Using Similarity-Constraints Deep Neural Networks. In *International Conference on Database Systems for Advanced Applications*. Springer, 587–603.

[64] Stijn J Rotman, Boris Cule, and Len Feremans. 2023. Efficiently Mining Frequent Representative Motifs in Large Collections of Time Series. In *2023 IEEE International Conference on Big Data (BigData)*. IEEE, 66–75.

[65] Lucie Saclova, Andrea Nemcova, Radovan Smisek, Lukas Smital, Martin Vitek, and Marina Ronzhina. 2022. Reliable P wave detection in pathological ECG signals. *Scientific Reports* 12, 1 (2022), 6589.

[66] Saquib Sarfraz, Naila Murray, Vivek Sharma, Ali Diba, Luc Van Gool, and Rainer Stiefelhagen. 2021. Temporally-weighted hierarchical clustering for unsupervised action segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 11225–11234.

[67] Patrick Schäfer and Ulf Leser. 2022. Motiflets: Simple and Accurate Detection of Motifs in Time Series. *Proceedings of the VLDB Endowment* 16, 4 (2022), 725–737.

[68] Sebastian Schmidl, Phillip Wenig, and Thorsten Papenbrock. 2022. Anomaly detection in time series: a comprehensive evaluation. *Proc. VLDB Endow.* 15, 9 (May 2022), 1779–1797. https://doi.org/10.14778/3538598.3538602

[69] Pavel Senin, Jessica Lin, Xing Wang, Tim Oates, Sunil Gandhi, Arnold P Boedihardjo, Crystal Chen, and Susan Frankenstein. 2018. Grammarviz 3.0: Interactive discovery of variable-length time series patterns. *ACM Transactions on Knowledge Discovery from Data (TKDD)* 12, 1 (2018), 1–28.

[70] Rodger Staden. 1989. Methods for discovering novel motifs in nucleic acid sequences. *Bioinformatics* 5, 4 (1989), 293–298.

[71] Zbigniew R Struzik and Arno Siebes. 1999. Measuring time series similarity through large singular features revealed with wavelet transformation. In *Proceedings. Tenth International Workshop on Database and Expert Systems Applications. DEXA 99*. IEEE, 162–166.

[72] Zeeshan Syed, Collin Stultz, Manolis Kellis, Piotr Indyk, and John Guttag. 2010. Motif discovery in physiological datasets: a methodology for inferring predictive elements. *ACM Transactions on Knowledge Discovery from Data (TKDD)* 4, 1 (2010), 1–23.

[73] Yoshiki Tanaka, Kazuhisa Iwamoto, and Kuniaki Uehara. 2005. Discovery of time-series motif from multi-dimensional data based on MDL principle. *Machine Learning* 58 (2005), 269–300.

[74] Yoshiki Tanaka and Kuniaki Uehara. 2003. Discover motifs in multi-dimensional time-series using the principal component analysis and the mdl principle. In *International Workshop on Machine Learning and Data Mining in Pattern Recognition*. Springer, 252–265.

[75] Heng Tang and Stephen Shaoyi Liao. 2008. Discovering original motifs with different lengths from time series. *Knowledge-Based Systems* 21, 7 (2008), 666–671.

[76] N Tatbul. 2018. Precision and Recall for Time Series. *arXiv preprint arXiv:1803.03639* (2018).

[77] Sahar Torkamani and Volker Lohweg. 2014. Identification of multi-scale motifs. In *ProcEEDings 24. Workshop comPutational intElligEncE*. 277.

[78] Sahar Torkamani, Volker Lohweg, F Hoffmann, and E Hüllermeier. 2015. Shift-invariant feature extraction for time-series motif discovery. In *25. Workshop Computational Intelligence, ser. Schriftenreihe des Instituts für Angewandte Informatik-Automatisierungstechnik am Karlsruher Institut für Technologie*, Vol. 54. 23–45.

[79] Cao Duy Truong and Duong Tuan Anh. 2015. A fast method for motif discovery in large time series database under dynamic time warping. In *Knowledge and Systems Engineering: Proceedings of the Sixth International Conference KSE 2014*. Springer, 155–167.

[80] Daan Van Wesenbeeck, Aras Yurtman, Wannes Meert, and Hendrik Blockeel. 2024. LoCoMotif: Discovering time-warped motifs in time series. *Data Mining and Knowledge Discovery* (2024), 1–30.

[81] Ugo Vespier, Siegfried Nijssen, and Arno Knobbe. 2013. Mining characteristic multi-scale motifs in sensor-based time series. In *Proceedings of the 22nd ACM international conference on information & knowledge management*. 2393–2398.

[82] Dragomir Yankov, Eamonn Keogh, Jose Medina, Bill Chiu, and Victor Zordan. 2007. Detecting time series motifs under uniform scaling. In *Proceedings of the 13th ACM SIGKDD international conference on Knowledge discovery and data mining*. 844–853.

[83] Chin-Chia Michael Yeh, Nickolas Kavantzas, and Eamonn Keogh. 2017. Matrix Profile VI: Meaningful Multidimensional Motif Discovery. In *2017 IEEE International Conference on Data Mining (ICDM)*. 565–574. https://doi.org/10.1109/ICDM.2017.66

[84] Chin-Chia Michael Yeh, Yan Zhu, Liudmila Ulanova, Nurjahan Begum, Yifei Ding, Hoang Anh Dau, Diego Furtado Silva, Abdullah Mueen, and Eamonn Keogh. 2016. Matrix profile I: all pairs similarity joins for time series: a unifying view that includes motifs, discords and shapelets. In *2016 IEEE 16th international conference on data mining (ICDM)*. Ieee, 1317–1322.

[85] Myat Su Yin, Songsri Tangsripairoj, and Benjarath Pupacdi. 2014. Variable length motif-based time series classification. In *Recent Advances in Information and Communication Technology: Proceedings of the 10th International Conference on Computing and Information Technology (IC2IT2014)*. Springer, 73–82.

[86] Sorrachai Yingchareonthawornchai, Haemwaan Sivaraks, Thanawin Rakthanmanon, and Chotirat Ann Ratanamahatana. 2013. Efficient proper length time series motif discovery. In *2013 IEEE 13th international conference on data mining*. IEEE, 1265–1270.

[87] Yan Zhu, Chin-Chia Michael Yeh, Zachary Zimmerman, Kaveh Kamgar, and Eamonn Keogh. 2018. Matrix profile XI: SCRIMP++: time series motif discovery at interactive speeds. In *2018 IEEE international conference on data mining (ICDM)*. IEEE, 837–846.

[88] Yan Zhu, Zachary Zimmerman, Nader Shakibay Senobari, Chin-Chia Michael Yeh, Gareth Funning, Abdullah Mueen, Philip Brisk, and Eamonn Keogh. 2016. Matrix profile ii: Exploiting a novel algorithm and gpus to break the one hundred million barrier for time series motifs and joins. In *2016 IEEE 16th international conference on data mining (ICDM)*. IEEE, 739–748.