Spatial-Time Motifs Discovery

Heraldo Borges, Murillo Dutra, Amin Bazaz, Rafaelli Coutinho, Fábio Perosi, Fábio Porto, Florent Masseglia, Esther Pacitti, Eduardo Ogasawara

Spatial-Time Motifs Discovery

Heraldo Borges [a], Murillo Dutra [a], Amin Bazaz [d], Rafaelli Coutinho [a], Fábio Perosi [c], Fábio Porto [b], Florent Masseglia [e,d], Esther Pacitti [e,d] and Eduardo Ogasawara [a,*]

[a] *CEFET/RJ - Federal Center for Technological Education of Rio de Janeiro, Rio de Janeiro, Brazil*
[b] *LNCC - National Laboratory for Scientific Computing, Petropolis, Brazil*
[c] *UFRJ - Federal University of Rio de Janeiro, Rio de Janeiro, Brazil*
[d] *University of Montpellier Montpellier, France*
[e] *INRIA, France*

Abstract. Discovering motifs in time series data has been widely explored. Various techniques have been developed to tackle this problem. However, when it comes to spatial-time series, a clear gap can be observed according to the literature review. This paper tackles such a gap by presenting an approach to discover and rank motifs in spatial-time series, denominated Combined Series Approach (CSA). CSA is based on partitioning the spatial-time series into blocks. Inside each block, subsequences of spatial-time series are combined in a way that hash-based motif discovery algorithm is applied. Motifs are validated according to both temporal and spatial constraints. Later, motifs are ranked according to their entropy, the number of occurrences, and the proximity of their occurrences. The approach was evaluated using both synthetic and seismic datasets. CSA outperforms traditional methods designed only for time series. CSA was also able to prioritize motifs that were meaningful both in the context of synthetic data and also according to seismic specialists.

Keywords: Motifs, Spatial-Time Series, Seismic

1. Introduction

Under the data deluge scenario, Data Scientists are pushed to provide new ways for efficiently collecting, storing, processing, and organizing a large amount of data [1]. We are immersed in a scenario with massive databases from many sources, types, and formats. However, such scenario opens a set of research opportunities involving knowledge discovery [2, 3]. In this context, many phenomena can be observed and organized as a sequence of observations in a timeline that can be modeled as a time series, enabling discoveries.

A relevant area that is being explored in time series analysis is finding patterns [4]. Patterns are sub-sequences of time series that are related to some special properties or behaviors [5]. A particular pattern that occurs a significant number of times in time series is denominated *motif* [6].

The discovery of motifs enables the understanding of some specific behaviors observed in time series, in many areas of knowledge, such as weather prediction [7], wind generation [8], image recognition [9], seismic amplitude [10], and computation biology (such as protein discovery) [11, 12]. A vast number of motifs discovery techniques, methods, and algorithms have been developed [13–16]. They include discovering motifs of a particular/variable length [17, 18], or without constraints (parameter-free algorithms) [19], or in multivariate time series [20, 21].

However, various important time-series phenomena present different behaviors when observed at points of space (for example, series collected by sensors and IoT) and are better modeled as spatial-time series, in which each time series is associated to a position in space. Under such scenarios, motifs might not be discovered when we restrict the analysis to the time-only dimension.

Consider, for example, the scenario where speed sensors are present in each corner of Manhattan to monitor vehicles speed. Imagine that a car accident occurs in one corner. Such occurrence decreases the vehicles speed

*Corresponding author. E-mail: eogasawara@ieee.org.

there. It is a punctual phenomenon that might not occur again, *i.e.*, this pattern is not observable using motif discovery techniques for univariate time series. However, due to the accident, congestions may occur in the nearby corners, decreasing vehicles speed. This pattern repeatedly occurs in the nearby corners, possibly with a time lag. It characterizes, intuitively, what we would call as a spatial-time motif. The challenge becomes to identify regions of space and time where the motifs are frequently observed. Finding patterns that are frequent in a constrained space and time, *i.e.*, finding spatial-time motifs, may enable us to comprehend how a phenomenon occurs.

This paper tackles such problem by presenting an approach to discover and rank motifs in spatial-time series, denominated Combined Series Approach (CSA). CSA partitions the dataset into space-time blocks. Subsequences of time series inside these blocks are combined into a single time series. Then, it applies a traditional motif discovery algorithm over the combined time series to discover them. Subsequences occurring above a spatial-time threshold are selected as motifs. Finally, motifs are ranked considering their entropy, their number of occurrences, and the proximity of their occurrences.

We have evaluated our approach using Synthetic and Seismic datasets. It was able to identify spatial-time motifs that could not be identified using the traditional approach. CSA was also able to prioritize motifs that were meaningful both in the context of synthetic data and also according to seismic specialists.

In addition to this introduction, this work is organized into more seven sections. Section 2 presents the background for time series data mining. It includes a brief literature review about motifs in time series and the main concepts and techniques that support the motif discovering processes. Section 3 presents the Related Works. Section 4 formalizes the problem definition. Section 5 presents the CSA algorithm to find motifs in spatial-time series. Section 6 presents a synthetic dataset showing how the CSA behaves. Section 7 describes and explains the experiments that were made using seismic dataset. Finally, section 8 concludes.

2. Background

In this section, we introduce some background for the data mining process of motif discovery.

2.1. Time series background

A **time series** t is an ordered sequence of values in time: $t =< t_1, t_2, \cdots, t_m >, t_i \in \mathbb{R}$, where each t_i is a value, $|t| = m$ is the number of elements in t, and t_m is the most recent value in t [3].

A **subsequence** is a continuous sample of a time series with a defined length. The p-th **subsequence** of size n in a time series t, represented as $seq_{n,p}(t)$, is an ordered sequence of values $< t_p, t_{p+1}, \ldots, t_{p+n-1} >$, where $|seq_{n,p}(t)| = n$ and $1 \leqslant p \leqslant |t| - n$. Subsequences enable the analysis of data samples to evaluate some local properties [22].

Sliding windows consist of exploring all possible subsequences of a time series [23, 24]. Sliding windows produce a set of subsequences of the same length. All **sliding windows** of size n for a time series t are a function $sw_n(t)$ that produces a matrix W of size $(|t| - n + 1)$ by n. Each line w_i in W is the i-th subsequence of size n from t. Given $W = sw_n(t), \forall w_i \in W, w_i = seq_{n,i}(t)$. This concept is widely used in time series analysis to make comparisons between subsequences to find their similarities [25].

Figure 1 depicts the application of the above definitions to a time series. The blue line represents a time series t, the red line represents a subsequence from the time series, and the green dashed lines are examples of some of the subsequences extracted from time series based on sliding windows.

A spatial-time series st can be described as a pair (t, p), such that a time series t with an associated position p [26]. The position can be its geographical coordinates or any other reference that can represent the place where data had been observed. If the position is a function of time, it is a trajectory spatial-time series. Otherwise, it is a permanent spatial-time series. In this work, we are interested in permanent spatial-time series. For the sake of simplicity, from now on, we are calling them spatial-time series.

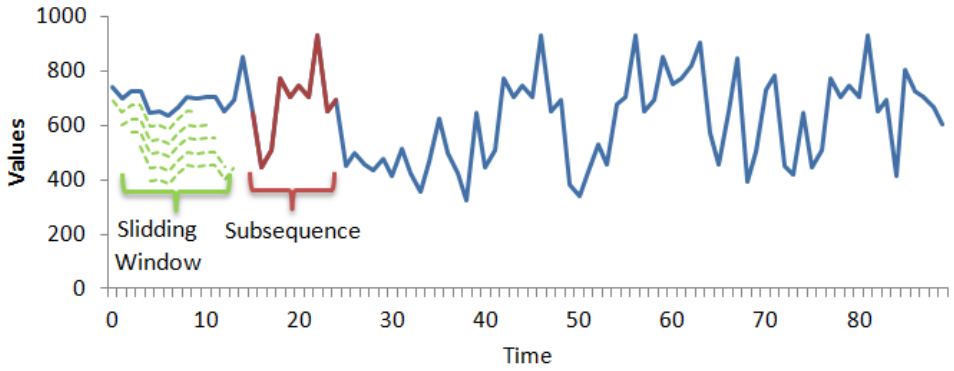


Fig. 1. An example of a time series, sub-sequences, and sliding windows

2.2. Data preprocessing

Data preprocessing techniques are key activities for enabling or improving the quality of data mining. In time series context, data is usually a continuous numerical value. For motif discovery, processing directly numerical representation is not efficient [27]. Due to that, during motif discovery, two data preprocessing techniques are commonly applied in sequence: (i) data normalization and (ii) Symbolic Aggregate Approximation (SAX).

Normalization is commonly used to enable the effectiveness of time series comparison methods. One of the most common normalization methods is z-score [28]. As a result of this normalization method, the normalized time series has zero as average and one as standard deviation. Equation 1 describes z-score normalization, where t_i is an observation of the time series t, μ_t is the average, σ_t is the standard deviation of the time series, and t' is the transformed time series. Additionally, the min-max is another normalization method that applies a linear transformation to the original data, where the minimum value ($min(t)$) and the maximum value ($max(t)$) are used to transform each value t_i to another value t'_i in a range varying from $[0, 1]$, as shown in Equation 2 [29].

$$t'_i = \frac{(t_i - \mu_t)}{\sigma_t} \tag{1}$$

$$t'_i = \frac{t_i - min(t)}{max(t) - min(t)} \tag{2}$$

SAX is an indexing technique. It consists primarily in partitioning the domain of a variable into ranges such that each range is associated with a particular symbol [30]. The SAX alphabet size defines the number of partitions for the domain. Thus, all values are replaced by their respective associated symbol. Given an alphabet $(a_1, \cdots, a_n)$ of size n, the values of time series t are divided into n ranges $([-\infty, \beta_1], \cdots, [\beta_{n-1}, \infty])$ according to Gaussian function (with different sizes, but same probability), such that each value t_i is mapped to an alphabet value a_k, where $1 \leqslant k \leqslant n$ [31].

2.3. Motif

Given a sequence q and time series t, q is a **motif** in t with support σ, if and only if q is included in t at least σ times. The length of a motif q ($|q|$) is also known as word size. Formally, given a sequence q and a time series t where $W = sw_{|q|}(t)$, $motif(q, t, \sigma) \leftrightarrow \exists\, R \subseteq W, (|R| \geqslant \sigma)$, such that $\forall w_i \in R, w_i = q$ [13]. An important property

regarding motifs is that the repeated subsequence is not previously known and is discovered when scanning the entire data. It can be discovered by making a comparison between subsequences that are obtained using sliding windows [22].

Many methods proposed in the literature to discover motifs in time series are computationally intensive [4]. Due to that, many works aim to improve the effectiveness of motif discovery and reduce the computational resources needed. Such a process requires some data preprocessing such as normalization and indexing before running the motif discovery algorithms to increase the performance and precision of results [13].

There are some main approaches to discover motifs, such as (i) brute force; (ii) heuristic-based; and (iii) matrix profile. The brute force approach is the simplest method. It has a high computational cost, especially when used to discover sequences of greater size in large datasets [32]. It is indicated for discovering sequences of smaller size [33]. In this method, the coverage and accuracy are complete since it makes all possible comparisons between all subsequences of a time series.

The heuristic-based algorithms include methods such as random projections. The goal is to reduce the search space. The random projection was proposed to handle large dataset by reducing dimensionality. It randomly selects some of sliding windows columns for search [33]. The technique optimizes the execution time in discovering motifs [22, 34]. It adopts a collision matrix that is built by masking the projected columns of both the subsequence matrix and candidate search sequence. If they match, then the sequence is placed in a hash structure for full comparison.

The matrix profile is based on computing the distance of a sequence of size n with the most similar subsequence present in the time series. It is called matrix profile since the naive implementation of this technique is to compute all pairwise distance for all sequences present in the time series. However, it can use efficient algorithms, such as Fast Fourier Transform (FFT) to enable very fast computation [16].

After motif discovering process, an important task in motif analysis is how to sort the motifs according to their relevance [35]. A standard classification method is k-motif which considers the total number of occurrences of the motifs in time series. Also, motifs can be sorted according to their relevance degree. For instance, some motifs can be similar to a straight line (*i.e.*, constant observations) and depending on the data domain may not be relevant. Such motifs can be low qualified or discarded to avoid distorting the analysis [22].

Some approaches to evaluate the significance and relevance of motifs were proposed in the literature. A statistical approach to assess the relevance of motifs is based on information gain, which measures how expected is the motif to occurs [35]. The Log-odds considers the degree of how rare the motif is by comparing the amount of occurrence with the expected chance of having occurrence based on probabilistic distribution [36]. Castro and Azevedo [35] proposed the estimation of expectation for the frequency of a motif based on Markov Chain models. The value is assessed making the comparison between actual frequency and estimated based on hypothesis tests.

3. Related Works

There are two recent review papers regarding motif discovery that characterize researches and trends [13, 15]. Concerning spatial-time approach, there are few initiatives such as discovering motifs in trajectory data [37] and discovering migration motifs in financial data [38]. Oates et al. [37] focused on analyzing repetitive sequences of moving objects. For that, they developed a grammar, applied SAX indexing, and searched for motifs over the trajectory. In our work, we do not have a moving object. Sensors are fixed, and we analyze a phenomenon that occurs at each position throughout time.

Meanwhile, in Du et al. [38], space is modeled by discrete attributes that resemble states of an object. In the context of their paper, they refer to the state of companies in the stock market. It is, in fact, a state-space model [3] where a trajectory is the registration of state transitions. In this way, it differs significantly from our work, since such a phenomenon may not be constrained in space and time.

Due to the absence of directly related work for the spatial-time motif discovery, for the sake of our work, we can group time series motif discovery approaches according to the exactness (exact or approximate), length (fixed or variable), and dimension (single or multiple).

Considering the exact motif discovery approach, some specific method to address the dimensionality and motif length problem were proposed. Jiang et al. [39] proposed an efficient motif discovery algorithm PMDGS (P-Motif Discovery based on Grid Structure) that processes data streams. Mueen et al. [32] proposed a motif discovery algorithm for exact time series called MK (Mueen-Keogh) and observed that MK was faster than brute-force [22]. Narang and Bhattacherjee [40] introduced the Par-MK, Par-MK-SLB, and Par-MK-DLB, which are all parallel multi-threaded algorithms for exact motif discovery that focus on load balancing. Mueen et al. [41] proposed a disk-aware algorithm to discover exact motifs in large time series databases. Cassisi et al. [10] applied a motif discovery technique for an exact time series to study recurrent patterns in seismic amplitude time series of the Etna 2011 periodic eruptive activity. Chi and Wang [42] introduced a method based on cloud model theory to extract the top k-motifs. Truong and Anh [43] proposed a fast method for motif discovery in time series based on Dynamic Time Warping distance.

When it comes to approximate motif discovery, the approaches aim to reduce the complexity and, consequently, the computational cost. Some work proposed approaches to improve the accuracy and efficiency of Random Projection Algorithm as proposed in Chiu et al. [22]. Lin et al. [30] created a new symbolic representation for time series (SAX) for indexing. Mohammad and Nishida [44] proposed two algorithms called MCFull and MCInc that address constrained motif discovery problem. Castro and Azevedo [45] addressed motif discovery problem as an approximate top k-frequent subsequence discovery problem. Lin et al. [46] presented an approach that uses subseries joins to get similarity among subseries of the time series. Armstrong and Drewniak [47] developed the algorithm MD-RP for unsupervised motif discovering in time series. Narang and Bhattcherjee [48] designed the new sequential and parallel Motif discovery and data deduplication algorithms based on bloom filters.

Regarding variable-length motif discovery, Wilson et al. [49] proposed the Motif Tracking Algorithm (MTA) that uses a small number of parameters based on the implementation of the Bell immune memory theory. Yankov et al. [50] presented a novel algorithm that discovers motifs in time series with invariance to uniform scaling. It enables to reduce parameters such as motif length. Nunthanid et al. [51] described the VLMD motif discovery algorithm that does not require the motif length parameter. Such an algorithm automatically returns motif lengths from all possible sliding window lengths, reducing a set of possibilities of the sliding window lengths. Nunthanid et al. [19] presented the k-Best Motif Discovery (kBMD) algorithm that is parameter-free. It produces a set of the best motif that is ranked by a scoring function based on similarity of motif locations and shapes. Mueen [52] proposed the MOEN, an exact free-parameter algorithm to enumerate motifs that is faster than brute-force approach due to a novel bound on the similarity function that uses only linear space. Mohammad and Nishida [53] proposed an extension of the MK algorithm called MK++ to handle multiple motifs of variable lengths considering maximum overlap of subsequences.

Finally, in multivariate time series, Tanaka and Uehara [54] and Tanaka et al. [55] showed how to dynamically determine the optimum period length using the Minimum Description Length (MDL) principle and applied the method to the multidimensional time-series transforming into one-dimensional time-series using the Principal Component Analysis. Liu et al. [56] proposed a heuristic approach that can significantly improve the quality of motifs in m-dimensional time series. Vahdatpour et al. [57] proposed a new model based on Random Projection to find approximately motif in multivariate time series data by combining motifs discovered and grouping them. Wang et al. [58] developed the AMG method to list the motifs by scanning the entire series and then filling a matrix for similarity comparison to verify the real motif. Lam et al. [59] proposed two algorithms for solving multivariate time series called nmotif and kmotif. McGovern et al. [7] introduced an approach to multidimensional motif mining in temporal streams of real-world data. Son and Anh [60] presented an R*-tree together with early abandoning based approach that stores Minimum Bounding Rectangles (MBR) of data in memory. Son and Anh [61] proposed two new algorithms for motif discovery in time series data, first based on R-tree and the other is based on dimensionality reduction through Skyline index.

4. Problem Definition

The motif discovery approaches presented in the literature review propose to solve the problem of discovering motifs on time series. In the context of spatial-time series, we observe a more complex scenario due to spatial

constraints. In order to highlight this challenging problem, consider a synthetic dataset containing twelve spatial-time series ($ST1 \cdots ST12$) as depicted in Figure 2. Each spatial-time series STi has a position in space. The ordering of time-series obeys their spatial placement. For example, $ST2$ is both close to $ST1$ and $ST3$.

If we apply to this scenario a known motif discovery method on each spatial-time series for a support $\sigma \geqslant 2$, we can observe discovered motifs in Figure 2, which are marked as green worm-like shape found only in $ST3$. Also, even when some motifs are discovered, considering the entire dataset, those motifs are not fully explored: there are many other equivalent worm-like shapes that are not discovered, although appearing in close spatial-time series ($ST2$, $ST4$). It is also possible to observe that similar subsequences appearing in neighboring spatial-time series are not discovered as motifs. They are depicted in Figure 5 as orange trapezium-like and red stripe-like motifs.

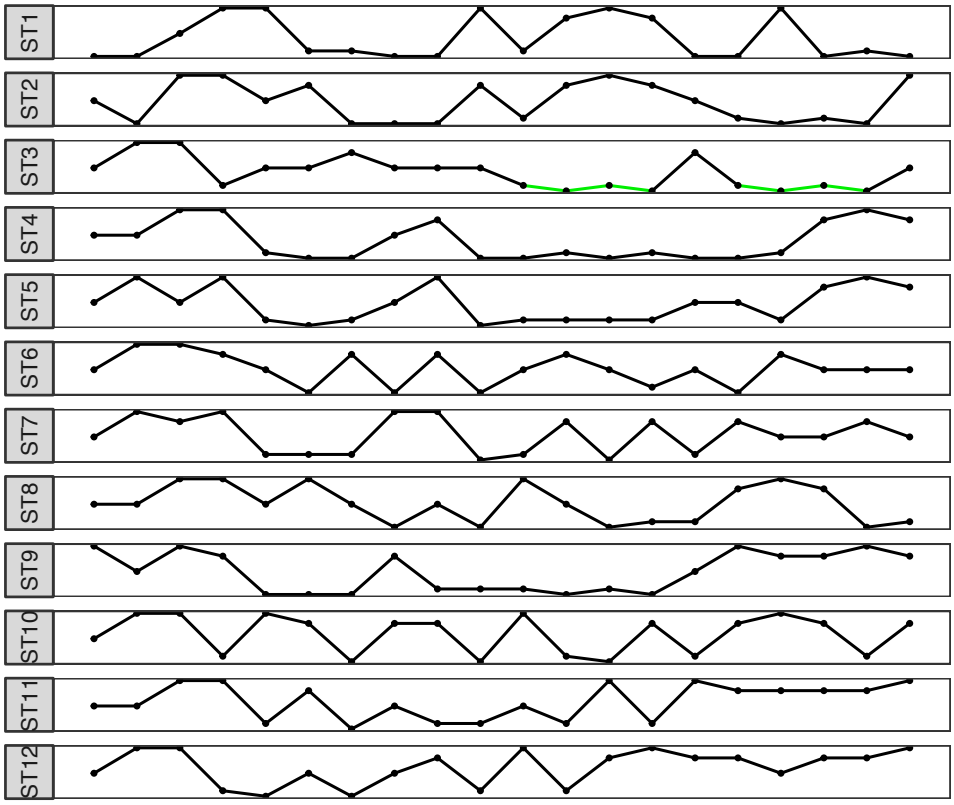


Fig. 2. A synthetic dataset with twelve spatial-time series: $ST1 \cdots ST12$. Traditional motif discovery algorithm applied in this spatial-time series dataset finds the two green worm-like as motifs

Depending on the dataset, such similar subsequences in neighboring time series can correspond to some relevant information. Discovering and grouping motifs in spatial-time series datasets can address some real-world problems. Such a scenario was not studied in previous works as discussed in section 3. The problem formalization for this new scenario is presented as follows.

A **spatial-time series dataset** (for short, dataset) S is a set of spatial-time series st. We are interested in finding motifs that occur in a constrained space and time. In our work, sequences may only be frequent inside spatial-time blocks. A **block** b is a couple $(\{st\}, i)$ where $\{st\}$ is a subset of neighboring spatial-time series and i is a time interval. The size of a block b is the product of the number of spatial-time series by the interval length: $|b| = |\{st\}| \cdot |i|$.

Let B be a partition of S into blocks b. Let σ and κ be two support values such that $\sigma \geqslant \kappa$. A subsequence q is a **spatial-time motif** if and only if there exists a block b such that q is included at least σ times in it and q occurs in at least κ different spatial-time series inside b.

From the definition above, the problem can be summarized as *the discovery of spatial-time motifs in a spatial-time series dataset*.

5. Combined Series Approach (CSA)

To address the defined problem, we developed the Combined Series Approach (CSA) that is organized in three main steps: (i) normalization & SAX indexing; (ii) discovery of spatial-time motifs; (iii) ranking of spatial-time motifs. CSA is summarized in Algorithm 1. It takes as input a spatial-time series dataset S, a word size w, an alphabet size a, sb and tb corresponding to spatial and temporal block sizes, and σ and κ constraints.

Algorithm 1 Combined Series Approach

function $CSA(S, w, a, sb, tb, \sigma, \kappa)$
  $S \leftarrow normSAX(S, a)$
  $stmotifs \leftarrow discoverSTMotifs(S, w, sb, tb, \sigma, \kappa)$
  $rstmotifs \leftarrow rankSTMotifs(stmotifs)$
  return $rstmotifs$

function $normSAX(S, a)$
  $S \leftarrow zscore(S)$
  $S \leftarrow SAX(S, a)$
  return S

function $discoverSTMotifs(S, w, sb, tb, \sigma, \kappa)$
  $B \leftarrow partition(S, sb, tb)$
  $stmotifs \leftarrow \emptyset$
  for each $b_{i,j} \in B$ **do**
    $cs \leftarrow combine(b_{i,j})$
    $motifs \leftarrow discover(cs, w, \sigma)$
    $stmotifs \leftarrow validate(motifs, \sigma, \kappa) \cup stmotifs$
  return $stmotifs$

function $rankSTMotifs(stmotifs)$
  $stmotifs \leftarrow group(stmotifs)$
  for each $m_i \in stmotifs$ **do**
    $ent_i = \sum_{k=1}^{|ft(m_i)|} \left(\frac{ft(m_i)_k}{n} \cdot log_2 \left(\frac{ft(m_i)_k}{n} \right) \right)$
    $O_i \leftarrow occurrences(m_i)$
    $occ_i \leftarrow log_2(O_i)$
    $prox_i \leftarrow \frac{1}{aw(mst(wam(O_i)))}$
  $rank = proj(norm(ent, occ, dist))$
  return $order(stmotifs, rank)$

5.1. Normalization & SAX indexing

The first step of the CSA, described by the $normSAX$ function of Algorithm 1, applies z-score data normalization in the entire dataset. Right after normalization, the SAX indexing method is applied for a given alphabet a. It

transforms the numeric data from S into letters according to the data distribution, as described in section 2.2. The output of such transformation is returned by the function.

5.2. Spatial-time Motif Discovery

The second step of CSA corresponds to $discoverSTMotifs$ function. In line 2, the indexed spatial-time series dataset S is partitioned into blocks (B). These blocks are created based on spatial block size (sb) and temporal block size (tb). The sb corresponds to the number of neighboring spatial-time series inside each block. The tb specifies the time interval for subsequences of spatial-time series. B is the partition of S into a set of blocks $\{b_{i,j}\}$. Formally, each block $b_{i,j}$ contains $sb \cdot tb$ observations, $\forall i \in [1, \frac{|st.t|}{tb}]$, $\forall j \in [1, \frac{|S|}{sb}]$. Each block $b_{i,j}$ contains sb subsequences q_k, such that $q_k = seq_{tb,(i-1)\cdot(tb)+1}(S_{(j-1)\cdot sb+k}.t)$, $\forall k \in [1, sb]$.

In line 5, all sequences inside a block are combined into a single time series cs, such that cs is the concatenation of sequences inside the block $b_{i,j}$. Formally, $cs = q_1 \| \cdots \| q_k$ and $|cs| = \sum_{i=1}^{k} |q_i| = sb \cdot tb$.

The $discover$ function (line 6) applies a traditional motif discovery approach. More specifically, $discover$ function applies an adapted hash-based approach [62] for exact match motif discovery [52][1]. It checks all subsequences of size w in cs to discovery motifs of length w. The first step applies a hash function for registering the positions of each subsequence s_i. If the number of occurrences of s_i is greater than σ, s_i is a temporal motif and included at $motifs$. Then, in line 7, $motifs$ are validated according to both σ and κ. It checks the number of distinct spatial-time series for them is greater or equal to κ. It is worth mentioning that any motif whose sequence appears distributed between neighboring subsequences of a block (for example, q_k and q_{k+1}) are fake occurrences and not considered during σ and κ validation. Motifs that validates both σ and κ are added at $stmotifs$.

5.3. Rank motifs

Since the number of motifs can be high, especially when working with larger alphabets, it is important to establish ways to rank them in a way that more "interesting" ones are presented first. The third step of CSA, described by the $rankSTMotifs$ function of Algorithm 1, makes a balance among three criteria: (i) the number of occurrences (significant higher occurrences are better); (ii) proximity (occurrences that are close to one another are better than the ones that are sparsely distributed in the dataset); (iii) entropy (higher entropy contains more information, which makes it more interesting).

Each motif corresponds to a sequence of SAX observations. All motifs that are discovered inside $discoverSTMotifs$ are local block motifs. At $group$ function (line 2), occurrences of motifs sharing the same sequence are grouped as long as they occur in neighboring blocks.

Then, for each group of motifs m_i, metrics for ranking them are computed. In line 4, the entropy of a motif m_i of size n is computed (ent_i). The ent_i is based on information theory and uses the frequency table (fm) for the characters presented in a motif [63] and is described in line 4. The higher is the entropy; the higher is the information that motif m_i encodes.

At line 5 the set of occurrences O_i for the motif m_i is obtained. Then, in line 6, the impact of the number of occurrences (O_i) of the i-th motif in a logarithm scale is computed (occ_i). This enables that only a significantly different number of occurrences becomes apparent.

In line 7, the weight of the occurrences according to their proximity is computed. Consider the pairs of position and time for the set of occurrences O_i of a motif m_i discovered in neighboring blocks. The distance between all these pairs is represented as a weighted adjacent matrix (wam). Then, the minimum spanning tree (mst) is built from the wam. Finally, the average weight (aw) for the edges of the mst is computed. Thus, $prox_i$ establishes the reciprocal measure for aw for the motif occurrences. The closer this measure is to 1, the closer are the occurrences in establishing a spatial-temporal pattern.

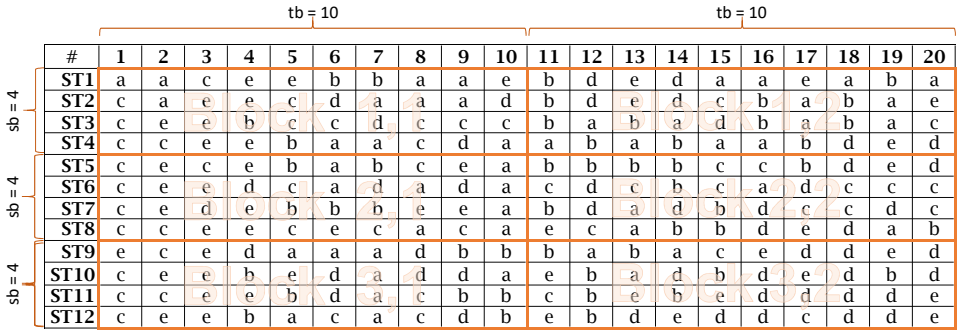
[1] Such an approach enables the introduction of other state-of-the-art motif discovery algorithms

Once the entropy (ent_i), the number of occurrences (occ_i), and the proximity ($prox_i$) has been computed for each motif; the ranking procedure can be applied. During ranking, each of these dimensions is normalized using min-max and projected into the unit vector that combines these three dimensions. Such projection provides a balance among these dimensions. The closer the projection of a motif m_i is to $(1,1,1)$, the better ranked it is. Function $rankSTMotifs$ returns the $stmotifs$ ordered according to the computed rank.

6. Analysis Using Synthetic Dataset

For a better understanding of the CSA and its steps, consider a synthetic spatial-time dataset S as depicted in Figure 2 (section 4). Each row is a spatial-time series (varying from positions 1 to 12) with 20 observations. In our example, we established the following thresholds: (i) word size ($w = 4$), (ii) alphabet size ($a = 5$), (iii) spatial block size ($sb = 4$), (iv) temporal block size ($tb = 10$), (v) thresholds $\sigma = 2$ and $\kappa = 2$.

Figure 3 depicts the result of applying the first step of CSA ($normSAX$) into the synthetic dataset. Values are being replaced by letters (a,b,c,d,e). In this case, central values are tagged as c, lower positive values as d, higher positive as e, lower negative as b and higher negative as a.



	#	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
sb = 4	ST1	a	a	c	e	e	b	b	a	a	e	b	d	e	d	a	a	e	a	b	a
	ST2	c	a	e	e	c	d	a	a	a	d	b	d	e	d	c	b	a	b	a	e
	ST3	c	e	e	b	c	c	d	c	c	c	b	a	b	a	d	b	a	b	a	c
	ST4	c	c	e	e	b	a	a	c	d	a	a	b	a	b	a	a	b	d	e	d
sb = 4	ST5	c	e	c	e	b	a	b	c	e	a	b	b	b	b	c	c	b	d	e	d
	ST6	c	e	e	d	c	a	d	a	d	a	c	d	c	b	c	a	d	c	c	c
	ST7	c	e	d	e	b	b	b	e	e	a	b	d	a	d	b	d	c	c	d	c
	ST8	c	c	e	e	c	e	c	a	c	a	e	c	a	b	b	d	e	d	a	b
sb = 4	ST9	e	c	e	d	a	a	a	d	b	b	b	a	b	a	c	e	d	d	e	d
	ST10	c	e	e	b	e	d	a	d	d	a	e	b	a	d	b	d	e	d	b	d
	ST11	c	c	e	e	b	d	a	c	b	b	c	b	e	b	e	d	d	d	d	e
	ST12	c	e	e	b	a	c	a	c	d	b	e	b	d	e	d	d	c	d	d	e

Fig. 3. Synthetic dataset partitioned into blocks

The second step of CSA ($discoverSTMotifs$) encompasses *combine*, *discover*, and *validate* functions. Figure 3 also shows the partitioning of the dataset according to CSA, where each orange box corresponds to a block. Since the dataset has 12 spatial-time series and 20 observations, the dataset is divided into 6 blocks, where each one contains 40 observations.

Figure 4 shows the result of the *combine* and *discover* functions presented in the $discoverSTMotifs$ to our synthetic dataset. Each block produces a combined time series (cs) with 40 observations. In each cs, the *discover* identifies all motif with $\sigma \geqslant 2$. The motifs discovered are marked with colors red, green, and orange. Then, motifs are depicted at their original position concerning the dataset (as presented in Figure 5) as long as they are validated according to both σ and κ constraints.

The majority of motifs discovered using the CSA approach had not been found when applying traditional motif discovery algorithms on each spatial-time series. As indicated in Table 1, traditional approach (Trad.) only found two occurrences for motif *baba* (marked as green). CSA found four occurrences of *baba*, seven of *bded* (marked as red), and two sets of three *ceeb* (marked as orange). In the case of *bded*, the seven occurrences were discovered in neighboring blocks and were grouped in a single set. However, in the case of *ceeb*, the two sets of three occurrences were not grouped since they were not found in neighboring blocks.

Table 1 also presents the dimensions used to rank the identified motifs. The entropy (Ent.) for both *bded* and *ceeb* was 1.5 since they contain three out of four distinct characters. The proximity metric (Prox.) is the reciprocal of the average weight of the minimum spanning tree that connects identified occurrences for each motif (the closer to one, the better it is). Thus, *baba* presents better proximity (0.83). The occurrences (Occ.) consider the log_2 of

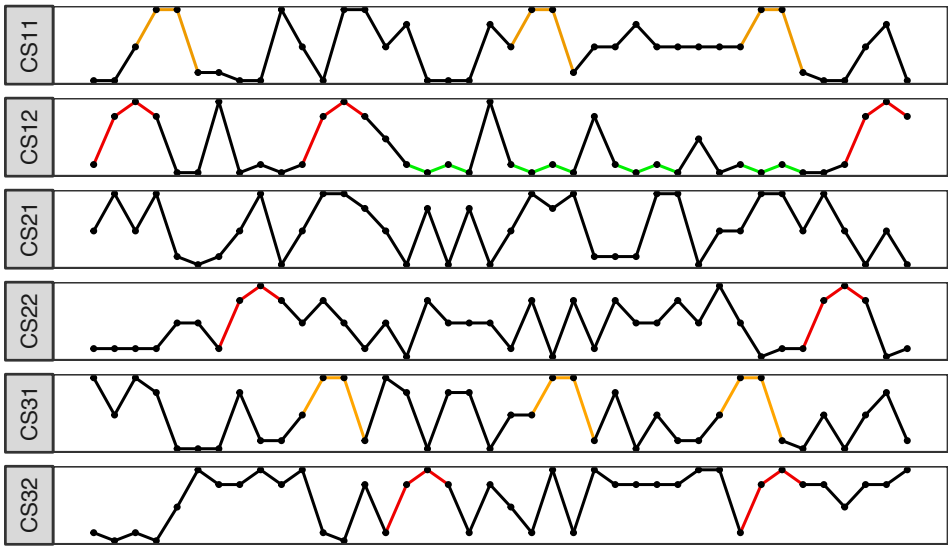


Fig. 4. Motif discovery algorithm applied to combined series

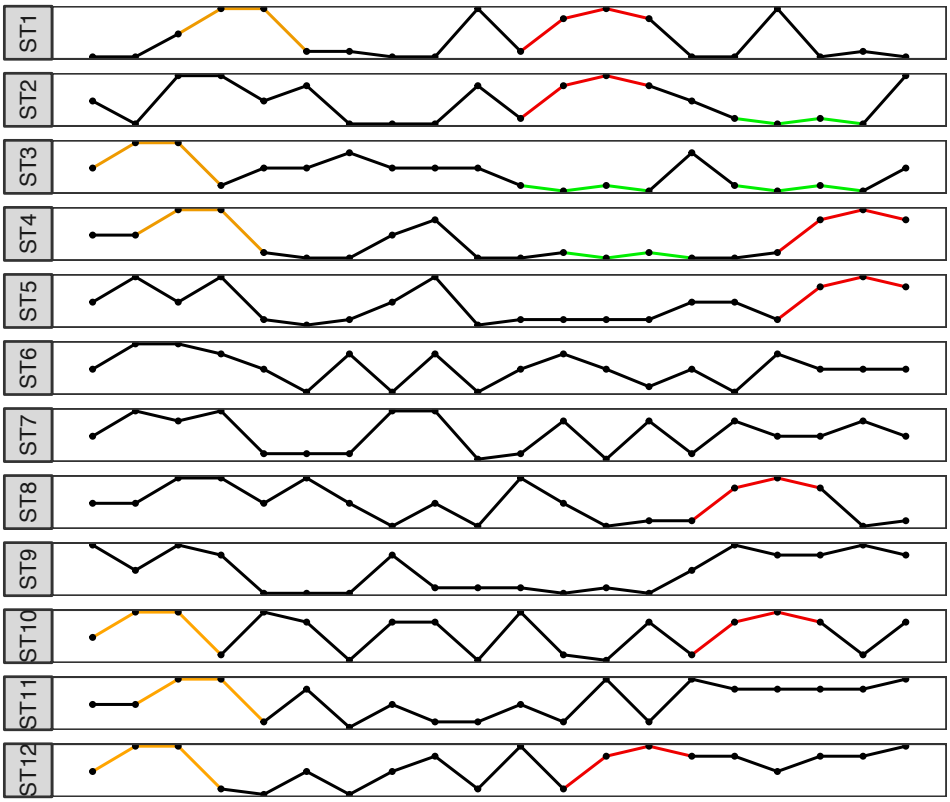


Fig. 5. Discovered motifs after spatial-temporal validation

the occurrences. Finally, the ranking (Rank) combines the normalized dimensions (Ent., Prox. and Occ.) projecting it to normalized vector $(\sqrt{\frac{1}{3}}, \sqrt{\frac{1}{3}}, \sqrt{\frac{1}{3}})$. The motif *bded* was the better ranked one (1.52) followed close by set $ceeb(1)$ that occurred from $ST10$ to $ST12$ (1.17). Although *ceeb* has better proximity, the number of occurrences of *bded* compensated this measure. Finally, even though *baba* had a lower entropy, it was better ranked than $ceeb(2)$, which occurred from $ST1$ to $ST4$, due to better occurrences and proximity values.

Table 1
CSA versus Traditional (Trad.) method in the synthetic dataset

Motif	Trad.	CSA	Ent.	Prox.	Occ.	Rank
bded	-	7	1.5	0.53	2.81	1.52
$ceeb(1)$	-	3	1.5	0.71	1.58	1.17
baba	2	4	1.0	0.83	2.00	0.95
$ceeb(2)$	-	3	1.5	0.47	1.58	0.71

7. Analysis Using Seismic Dataset

As a proof of concept evaluation, we applied CSA on the Netherlands seismic spatial-time series dataset, named F3 Block [64]. The database produced by the seismic reflection method was collected in a region located in the Dutch sector of the North Sea. This method consists of generating artificial seismic waves with energy sources that disturb the medium, such as explosives or air guns (called seismic shots) and record the waveforms of the various interfaces in the subsoil using sensors (geophones or hydrophones), in that acquisition air guns and hydrophones were used. The generated wave propagates through the interior of the Earth and the Sea. The partially reflected waves are used to find interfaces between layers that have significant contrasting elastic properties. The time of arrival of each reflection is related to the propagation velocities of the seismic wave in each layer. In a first approximation, the recorded amplitude is related to the contrast of the acoustic impedance, a product of velocity and density of the layers that define the interface. This method is analogous to imaging the human body using ultrasound. However, unlike medicine, where the density contrasts are imaged, on seismic exploration, the effect of the acoustic impedance difference is more studied [65].

In F3 Block dataset, each spatial-time series has a position in which the hydrophone is placed. The dataset is organized into inlines (direction of the ship navigation). We selected the inline 401 since it has been mapped by seismic specialists who have annotated some relevant information. Figure 6 shows the inline 401. Inline 401 consists of 920 spatial-time series with 440 observations in each. The horizontal axis represents the position of the receivers and vertical axis represents the time, which is also related to the depth at the subsoil.

The value of observations represents the wave amplitude reflected from the subsoil at a particular position and depth. Figure 7 depicts the probability density function (PDF), where it is possible to observe a frequency distribution with a high concentration of values close to zero and with values mainly varying from -10000 to 10000. Also, the data available is free from noise and missing values. In contrast to the synthetic dataset, depicted in Figure 3, in this dataset spatial-time series are displayed vertically.

7.1. Experimental Setup

This section discusses the experimental setup aiming to evaluate CSA in discovering spatial-time motifs in the seismic dataset. The setup was driven for a sensitivity analysis to measure CSA performance under different aspects and also against the traditional approach designed only for time series.

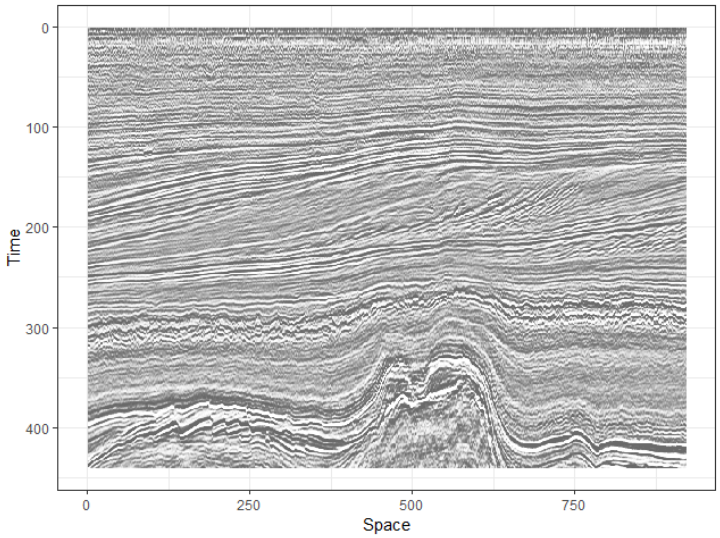


Fig. 6. Seismic Dataset

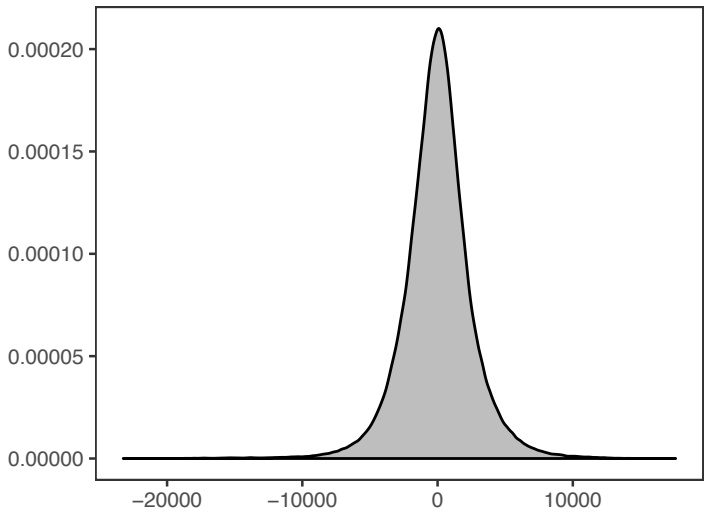


Fig. 7. Histogram of inline 401

The CSA algorithm requires parameters *alpha*, *word*, *tb*, *sb*, σ, and κ to be specified. The description of these parameters and the range explored are summarized in Table 2. These parameters influence both the quality of results and the computation elapsed time. The *alpha* was chosen based on the data adjustment. We varied the alphabet size for SAX encoding from 1 to 25, and measured the Mean Squared Error (MSE) for each observation concerning the mean of each SAX character. The higher the alphabet size, the lower is the MSE. The choice for the alphabet was identified by the maximum curvature analysis as depicted in Figure 8. The point where the maximum curvature is achieved (in red) indicates that increasing more the alphabet does not bring much more benefit concerning the MSE.

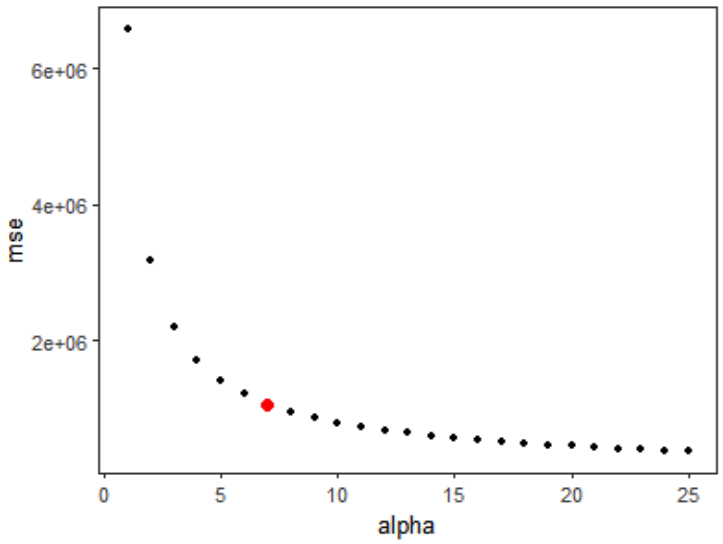


Fig. 8. *MS E* for each alphabet size

Table 2
Input Parameters

Parameter	Description (explored values)
alpha	Size of the alphabet for SAX indexing (fixed at 7)
word	Length of motif word (from 3 to 7)
tb x *sb*	Temporal and spatial block size (40x10, 20x20, 10x40)
σ	Minimum number of occurrences inside each block (from 2 to 7)
κ	Minimum number of spatial-time series with occurrences inside each block (from 1 to 5)

The CSA is available as an R Package (STMotif)[2]. The implementation, dataset, and results are also available[3]. All experimental evaluation was conducted in a cluster with 24 cores using SparkR [66]. The experimental evaluation ran at wall-time of 1.3 hours.

7.2. Analysis of Spatial-Time Motifs

In this analysis, the goal is to study the number of discovered motifs and their occurrences and computational time as we vary block size (*tb* and *sb*), *word*, σ, and κ.

To evaluate the influence of block orientation, we set three orientations: vertical rectangle (*tb* = 40; *sb* = 10), square (*tb* = 20; *sb* = 20), and horizontal rectangle (*tb* = 10; *sb* = 40). For a fair comparison, all of them contain the same amount of observations. We also included the traditional approach for discovering motifs in time series against CSA.

Table 3 presents the overall performance of both the traditional approach and CSA under different block orientation for all possible parameter combinations described in Table 2. The motifs column corresponds to the mean

[2]https://cran.r-project.org/web/packages/STMotif/index.html
[3]https://eic.cefet-rj.br/~eogasawara/csa

number of motifs whose occurrences were grouped with at least one neighboring block and contained more than seven occurrences (the maximum σ value adopted). In the case of the traditional approach, we considered it as a block of 440 temporal observations with one spatial-time series, so that the same grouping criteria could also be applied.

Table 3
Overall performance of CSA under different block orientation

Block orientation	motifs	sets of occur.	discovery time (min)	ranking time (min)	elapsed time (min)
Traditional (440x1)	43	449	1.8	2.0	4.7
CSA Vertical (40x10)	85	673	1.6	1.8	4.2
CSA Square (20x20)	114	772	1.4	1.6	3.8
CSA Horizontal (10x40)	105	705	0.9	1.2	2.9

The traditional approach, on average, discovered 43 different motifs under 449 sets of occurrences. It means that the same motif contains, on average, ten different spatial-temporal sets of occurrences. Also, the average discovery time and the average time to rank motifs were, respectively, 1.8 and 2.0 minutes. The average elapsed time was 4.7 minutes, which also includes the time to do the data normalization and SAX encoding.

The time for discovering motifs was approximately the same for all configuration, except for Horizontal orientation. In this setup, as we increase the size of the word, there is a lower number of possible motifs to discover. It becomes unnecessary to check for motifs in between two consecutive spatial-time series. It makes less possible comparisons for this setup, also meaning that a lower number of motifs are discovered. However, all CSA block orientations discovered more motifs than the traditional approach (the square had better performance. It discovered more than 2.5 times more motifs than traditional approach).

Comparing the performance of different CSA orientation (Vertical, Square, and Horizontal), we may expect that typically Horizontal orientation might break temporal sequences. However, in our dataset, patterns often occurred in a small time interval spread in space. Such behavior justifies the better performance of Horizontal orientation over Vertical one. Additionally, Square orientation had a better balance between time and space and was able to identify more patterns. The choice of block orientation is fairly domain-dependent. Users may consider their knowledge about the data to set up this parameter.

Table 4 disclosures the results of Table 3 according to the word size. It presents the number of discovered motifs and the sets of occurrences, applying the same criteria used to produce Table 3. It can be observed that as we increase the word size, the number of discovered motifs decreases. The same behavior occurs in the set of occurrences. The highest number of identified motifs occurred in CSA Square orientation for word size equals to four. Finally, the computation time (in minutes) for all discovered motifs also decreases as we increase the word size. It is due to the ranking function overhead. It has less impact on time when handling a lower number of occurrences.

Table 5 presents the influence of σ and κ in the number of discovered motifs for the CSA according to the CSA Square block orientation for word size equals to four. It is possible to observe that as we increase σ, lower number of occurrences are identified. Also, as we increase κ constraint, the number of occurrences decreases.

7.3. Analysis of Top-k Spatial-Time Motifs

Finally, we analyzed the top-k spatial-time motifs discovered using CSA Square block orientation for word size of four, fixing σ equals to three and κ equals to three. In this configuration, as presented in Table 6, we computed the top-5 distinct motifs that accomplished the same criteria adopted to build Table 3.

Table 4
Summary of Discovered Spatial-Time Motifs for different block orientation and word size

Block orientation	*word*	motifs	sets of occurrences	total time (min)
Traditional	3	139	95862	9.5
	4	65	6809	5.4
	5	7	369	3.0
	6	2	72	2.7
	7	1	17	2.6
CSA Vertical (40x10)	3	168	62278	8.0
	4	163	13980	4.7
	5	60	2988	3.3
	6	23	761	2.7
	7	11	229	2.5
CSA Square (20x20)	3	184	62324	6.7
	4	221	16887	4.5
	5	103	4182	3.1
	6	42	1157	2.4
	7	19	352	2.1
CSA Horizontal (10x40)	3	187	52199	5.5
	4	219	12901	3.7
	5	89	2918	2.3
	6	25	628	1.6
	7	7	149	1.2

Table 5
Influence of σ and κ in the number of occurrences in Square (20x20) setup with word size $w = 4$

	σ					
κ	2	3	4	5	6	7
1	42725	30052	21349	13559	9621	6959
2	42253	29938	21297	13527	9589	6927
3	-	29640	21191	13461	9530	6895
4	-	-	20073	13184	9368	6758
5	-	-	-	11900	8800	6490

The highest-ranked motif (*aagg*) presented good proximity value, an average entropy value, and a high occurrences value. Such combination produced a rank value of 1.57. The second place ($dfge$), although exhibiting low occurrences value, has a good proximity and entropy values. The third place (*aaag*) was similar to the first motif, but with lower occurrences value. The fourth place ($ggfa$) compensated the low occurrences value with an excellent proximity value. Finally, the fifth place ($egfa$) is similar to the second, with a slightly lower proximity value.

In order to have an intuition on the quality of the ranked motifs, we have plotted the top-ten discovered motifs (Figure 9), according to the ranking function, on top of the seismic dataset. The places where the motifs were

Table 6
Top five distinct motifs

motif	proximity	entropy	occurrences	rank
aagg	0.74	1.0	8.28	1.57
dfge	0.83	2.0	3.16	1.46
aaag	0.85	0.8	7.06	1.45
ggfa	1.00	1.5	3.17	1.40
egfa	0.75	2.0	3.17	1.39

plotted are in agreement with annotations from specialists where seismic horizons are located. Also, the yellow ones are very close to a gas reservoir.

In a complementary analysis, we sorted the motifs according to the number of occurrences. Figure 10 plots the top-ten distinct motifs sorted by their occurrences. The set of occurrences for each motif was plotted, as long as their ranking value were greater than 1.0. It can be observed that the occurrences of motifs matched more regions where seismic horizons are located.

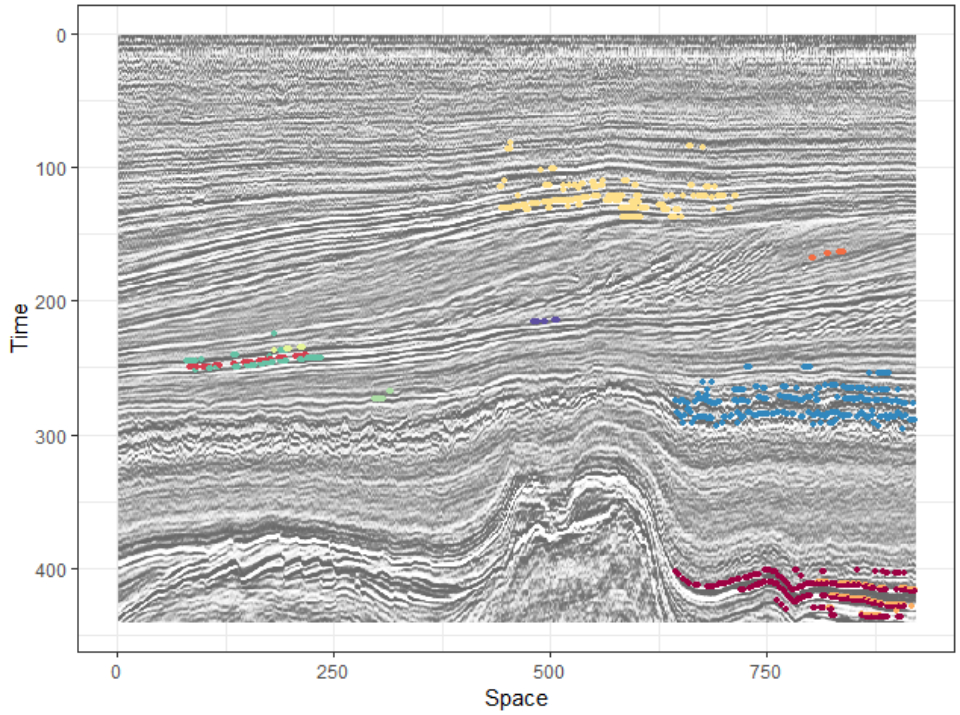


Fig. 9. Top-ten discovered motifs according to the ranking function

It is worth mentioning that the ranking function was conceived for general purpose usage and did not focus on any aspect to target seismic horizons. Nevertheless, they were able to discover the majority areas in which seismic horizons were annotated. Finally, our algorithm is capable of identifying more or fewer motifs according to the used parameters. However, the meaning of the motifs and their relevance is up to the specialist evaluation.

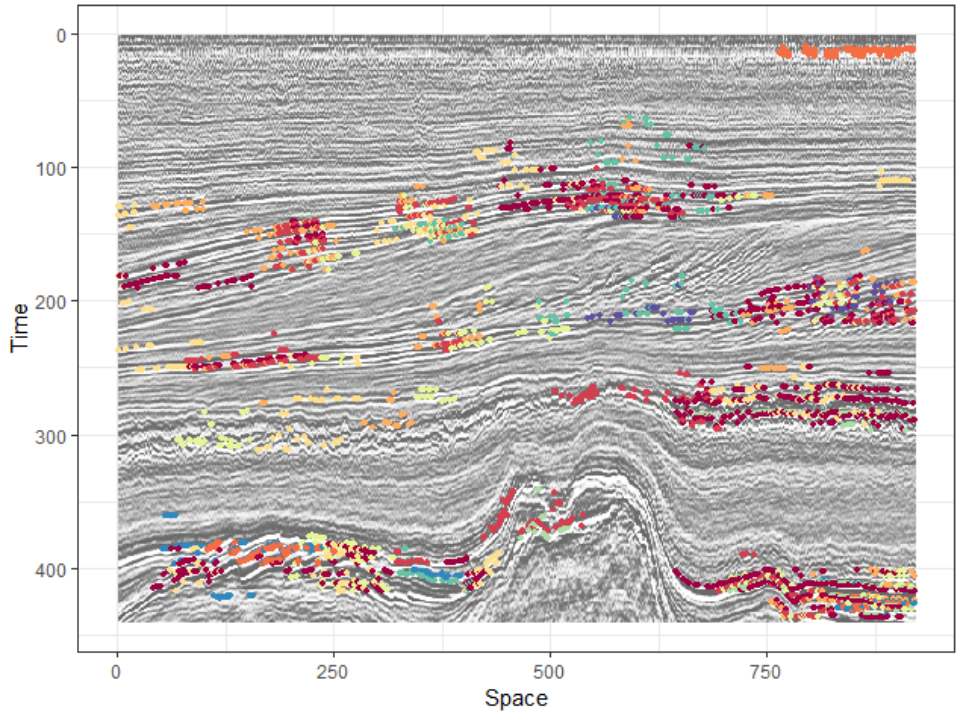


Fig. 10. Top-ten discovered motifs according to the number of occurrences, filtering the ones with ranking function lower than 1.0

8. Conclusion

Many applications observe phenomena whose values vary according to space and time dimensions. Discovering phenomena which are dependent on the occurrence in space and time requires extensions to traditional techniques adopted in time series analysis. In this paper, we tackle this problem by introducing a novel approach for spatial-time series motif discovery. To the best of our knowledge, this is the first work to propose a complete approach, named Combined Series Approach (CSA), for spatial-time motif discovery.

CSA supersedes traditional techniques when discovering spatial-time motifs, as it has been shown in our experimental evaluation. Additionally, CSA exhibits two major strengths points. Firstly, it is a divide-and-conquer algorithm that starts by discovering motifs inside a given spatial-time block. These blocks are then merged if neighboring blocks increase the number of occurrences of the discovered motifs. Such a technique makes the algorithm resilient to the initial block selection. Secondly, once the blocks have been defined, the approach is isolated from actual motif discovery algorithms applied. Such property enables exploring different motif discovery algorithms, such as Random Projection and Matrix Profile, exploring the effectiveness (precision), efficiency, and scalability targeting the improvement of space-time series discovery.

We have evaluated CSA against the traditional approach using both synthetic and seismic dataset. CSA was able to identify more motifs and occurrences than the traditional approach. Also, the identified motifs were well ranked considering both spatial-time constraints, number of occurrences and the motif entropy. Due to the potential of the technique applied to a spatial-time series, it opens opportunities for exploring other real-world applications modeled as spatial-time series. There are also opportunities to automate parameters for CSA. Finally, there are also opportunities to explore different ranking functions for targeting domain-specific problems.

Acknowledgements

The authors thank CNPq, CAPES (finance code 001), and FAPERJ for partially funding this research.

References

[1] C.-W. Tsai, C.-F. Lai, H.-C. Chao, and A.V. Vasilakos. Big data analytics: a survey. *Journal of Big Data*, 2(1), 2015.
[2] Jiawei Han, Micheline Kamber, and Jian Pei. *Data Mining: Concepts and Techniques*. Morgan Kaufmann, Haryana, India; Burlington, MA, 3 edition, July 2011. ISBN 978-0-12-381479-1.
[3] Robert H. Shumway and David S. Stoffer. *Time Series Analysis and Its Applications: With R Examples*. Springer, New York, NY, 4 edition, April 2017. ISBN 978-3-319-52451-1.
[4] P. Patel, E. Keogh, J. Lin, and S. Lonardi. Mining motifs in massive time series databases. In *Proceedings - IEEE International Conference on Data Mining, ICDM*, pages 370–377, 2002.
[5] J. Han, H. Cheng, D. Xin, and X. Yan. Frequent pattern mining: Current status and future directions. *Data Mining and Knowledge Discovery*, 15(1):55–86, 2007.
[6] E. Keogh and S. Kasetty. On the Need for Time Series Data Mining Benchmarks: A Survey and Empirical Demonstration. *Data Mining and Knowledge Discovery*, 7(4):349–371, 2003.
[7] A. McGovern, D.H. Rosendahl, R.A. Brown, and K.K. Droegemeier. Identifying predictive multi-dimensional time series motifs: An application to severe weather prediction. *Data Mining and Knowledge Discovery*, 22(1-2):232–258, 2011.
[8] Y.J. Fan and C. Kamath. Identifying and exploiting diurnal motifs in wind generation time series data. *International Journal of Pattern Recognition and Artificial Intelligence*, 29(2), 2015.
[9] L. Chi, Y. Feng, H. Chi, and Y. Huang. Face image recognition based on time series motif discovery. In *Proceedings - 2012 IEEE International Conference on Granular Computing, GrC 2012*, pages 72–77, 2012.
[10] C. Cassisi, M. Aliotta, A. Cannata, P. Montalto, D. Patanè, A. Pulvirenti, and L. Spampinato. Motif Discovery on Seismic Amplitude Time Series: The Case Study of Mt Etna 2011 Eruptive Activity. *Pure and Applied Geophysics*, 170(4):529–545, 2013.
[11] G.Z. Hertz and G.D. Stormo. Identifying DNA and protein patterns with statistically significant alignments of multiple sequences. *Bioinformatics*, 15(7-8):563–577, 1999.
[12] Jessica Lin, Eamonn Keogh, Stefano Lonardi, and Pranav Patel. Finding Motifs in Time Series. *Proceedings of the Second Workshop on Temporal Data Mining*, 2002.
[13] A. Mueen. Time series motif discovery: Dimensions and applications. *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*, 4(2):152–159, 2014.
[14] J. Serrà and J.L. Arcos. Particle swarm optimization for time series motif discovery. *Knowledge-Based Systems*, 92:127–137, 2016.
[15] S. Torkamani and V. Lohweg. Survey on time series motif discovery. *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*, 7(2), 2017.
[16] C.-C.M. Yeh, Y. Zhu, L. Ulanova, N. Begum, Y. Ding, H.A. Dau, Z. Zimmerman, D.F. Silva, A. Mueen, and E. Keogh. Time series joins, motifs, discords and shapelets: a unifying view that exploits the matrix profile. *Data Mining and Knowledge Discovery*, 32(1):83–123, 2018.
[17] Y. Li and J. Lin. Approximate variable-length time series motif discovery using grammar inference. In *Proceedings of the 10th International Workshop on Multimedia Data Mining, MDMKDD '10*, 2010.
[18] H. Tang and S.S. Liao. Discovering original motifs with different lengths from time series. *Knowledge-Based Systems*, 21(7):666–671, 2008.
[19] P. Nunthanid, V. Niennattrakul, and C.A. Ratanamahatana. Parameter-free motif discovery for time series data. In *2012 9th International Conference on Electrical Engineering/Electronics, Computer, Telecommunications and Information Technology, ECTI-CON 2012*, 2012.
[20] D. Minnen, C.L. Isbell, I. Essa, and T. Starner. Discovering multivariate motifs using subsequence density estimation and greedy mixture learning. In *Proceedings of the National Conference on Artificial Intelligence*, volume 1, pages 615–620, 2007.
[21] C.-C.M. Yeh, N. Kavantzas, and E. Keogh. Matrix profile VI: Meaningful multidimensional motif discovery. In *Proceedings - IEEE International Conference on Data Mining, ICDM*, volume 2017-November, pages 565–574, 2017.
[22] B. Chiu, E. Keogh, and S. Lonardi. Probabilistic discovery of time series motifs. In *Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 493–498, 2003.
[23] E. Keogh and J. Lin. Clustering of time-series subsequences is meaningless: Implications for previous and future research. *Knowledge and Information Systems*, 8(2):154–177, 2005.
[24] C.H. Lampert, M.B. Blaschko, and T. Hofmann. Beyond sliding windows: Object localization by efficient subwindow search. In *26th IEEE Conference on Computer Vision and Pattern Recognition, CVPR*, 2008.
[25] M. Van Hoan and M. Exbrayat. Time series symbolization and search for frequent patterns. In *ACM International Conference Proceeding Series*, pages 108–117, 2013.
[26] S. Shekhar, S.K. Feiner, and W.G. Aref. Spatial computing. *Communications of the ACM*, 59(1):72–81, 2016.

[27] C.S. Daw, C.E.A. Finney, and E.R. Tracy. A review of symbolic analysis of experimental data. *Review of Scientific Instruments*, 74(2): 915–930, 2003.

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

[29] E. Ogasawara, L. Murta, G. Zimbrão, and M. Mattoso. Neural networks cartridges for data mining on time series. In *Proceedings of the International Joint Conference on Neural Networks*, pages 2302–2309, 2009.

[30] J. Lin, E. Keogh, L. Wei, and S. Lonardi. Experiencing SAX: A novel symbolic representation of time series. *Data Mining and Knowledge Discovery*, 15(2):107–144, 2007.

[31] J. Lin, E. Keogh, S. Lonardi, and B. Chiu. A symbolic representation of time series, with implications for streaming algorithms. In *Proceedings of the 8th ACM SIGMOD Workshop on Research Issues in Data Mining and Knowledge Discovery, DMKD '03*, pages 2–11, 2003.

[32] A. Mueen, E. Keogh, Q. Zhu, S. Cash, and B. Westover. Exact discovery of time series motifs. In *Society for Industrial and Applied Mathematics - 9th SIAM International Conference on Data Mining 2009, Proceedings in Applied Mathematics*, volume 1, pages 469–480, 2009.

[33] L. Li and S. Nallela. Probabilistic discovery of motifs in water level. In *2009 IEEE International Conference on Information Reuse and Integration, IRI 2009*, pages 388–393, 2009.

[34] J. Buhler and M. Tompa. Finding motifs using random projections. *Journal of Computational Biology*, 9(2):225–242, 2002.

[35] N.C. Castro and P.J. Azevedo. Significant motifs in time series. *Statistical Analysis and Data Mining*, 5(1):35–53, 2012.

[36] J. Yang, W. Wang, and P.S. Yu. Mining surprising periodic patterns. *Data Mining and Knowledge Discovery*, 9(2):189–216, 2004.

[37] T. Oates, A.P. Boedihardjo, J. Lin, C. Chen, S. Frankenstein, and S. Gandhi. Motif discovery in spatial trajectories using grammar inference. In *International Conference on Information and Knowledge Management, Proceedings*, pages 1465–1468, 2013.

[38] X. Du, R. Jin, L. Ding, V.E. Lee, and J.H. Thornton Jr. Migration motif: A spatial-temporal pattern mining approach for financial markets. In *Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 1135–1143, 2009.

[39] T. Jiang, Y. Feng, B. Zhang, J. Shi, and Y. Wang. Finding motifs of financial data streams in real time. *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 5370 LNCS:546–555, 2008.

[40] A. Narang and S. Bhattacherjee. Parallel exact time series motif discovery. *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 6272 LNCS(PART 2):304–315, 2010.

[41] A. Mueen, E. Keogh, Q. Zhu, S.S. Cash, M.B. Westover, and N. Bigdely-Shamlo. A disk-aware algorithm for time series motif discovery. *Data Mining and Knowledge Discovery*, 22(1-2):73–105, 2011.

[42] H. Chi and S. Wang. Finding time series motifs based on cloud model. In *Proceedings - 2013 IEEE International Conference on Granular Computing, GrC 2013*, pages 70–75, 2013.

[43] C.D. Truong and D.T. Anh. A fast method for motif discovery in large time series database under dynamic time warping. *Advances in Intelligent Systems and Computing*, 326:155–167, 2015.

[44] Y. Mohammad and T. Nishida. Constrained motif discovery in time series. *New Generation Computing*, 27(4):319–346, 2009.

[45] N. Castro and P. Azevedo. Multiresolution motif discovery in time series. In *Proceedings of the 10th SIAM International Conference on Data Mining, SDM 2010*, pages 665–676, 2010.

[46] Y. Lin, M.D. McCool, and A.A. Ghorbani. Motif and anomaly discovery of time series based on subseries join. In *Proceedings of the International MultiConference of Engineers and Computer Scientists 2010, IMECS 2010*, pages 481–486, 2010.

[47] T. Armstrong and E. Drewniak. Unsupervised discovery of motifs under amplitude scaling and shifting in time series databases. *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 6871 LNAI: 539–552, 2011.

[48] A. Narang and S. Bhattcherjee. Real-time approximate range motif discovery & data redundancy removal algorithm. In *ACM International Conference Proceeding Series*, pages 485–496, 2011.

[49] W. Wilson, P. Birkin, and U. Aickelin. The motif tracking algorithm. *International Journal of Automation and Computing*, 5(1):32–44, 2008.

[50] D. Yankov, E. Keogh, J. Medina, B. Chiu, and V. Zordan. Detecting time series motifs under uniform scaling. In *Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 844–853, 2007.

[51] P. Nunthanid, V. Niennattrakul, and C.A. Ratanamahatana. Discovery of variable length time series motif. In *ECTI-CON 2011 - 8th Electrical Engineering/ Electronics, Computer, Telecommunications and Information Technology (ECTI) Association of Thailand - Conference 2011*, pages 472–475, 2011.

[52] A. Mueen. Enumeration of time series motifs of all lengths. In *Proceedings - IEEE International Conference on Data Mining, ICDM*, pages 547–556, 2013.

[53] Y. Mohammad and T. Nishida. Exact discovery of length-range motifs. *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 8398 LNAI(PART 2):23–32, 2014.

[54] Y. Tanaka and K. Uehara. Discover motifs in multi-dimensional time-series using the principal component analysis and the MDL principle. In *Lecture Notes in Artificial Intelligence (Subseries of Lecture Notes in Computer Science)*, volume 2734, pages 252–265, 2003.

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

[56] Z. Liu, J.X. Yu, X. Lin, H. Lu, and W. Wang. Locating motifs in time-series data. *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 3518 LNAI:343–353, 2005.

[57] A. Vahdatpour, N. Amini, and M. Sarrafzadeh. Toward unsupervised activity discovery using multi-dimensional motif detection in time series. In *IJCAI International Joint Conference on Artificial Intelligence*, pages 1261–1266, 2009.

[58] L. Wang, E.S. Chng, and H. Li. A tree-construction search approach for multivariate time series motifs discovery. *Pattern Recognition Letters*, 31(9):869–875, 2010.

[59] H.T. Lam, N.D. Pham, and T. Calders. Online discovery of top-k similar motifs in time series data. In *Proceedings of the 11th SIAM International Conference on Data Mining, SDM 2011*, pages 1004–1015, 2011.

[60] N.T. Son and D.T. Anh. Discovering time series motifs based on multidimensional index and early abandoning. *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 7653 LNAI(PART 1):72–82, 2012.

[61] N.T. Son and D.T. Anh. Discovery of time series k-motifs based on multidimensional index. *Knowledge and Information Systems*, 46(1): 59–86, 2016.

[62] P.T. Xuan and D.T. Anh. An efficient hash-based method for time series motif discovery. In *Multi-disciplinary Trends in Artificial Intelligence*, volume 11248 LNAI, pages 205–211, 2018.

[63] C.E. Shannon. A Mathematical Theory of Communication. *Bell System Technical Journal*, 27(3):379–423, 1948.

[64] dgbes. Netherlands Offshore F3 Block - Complete. Technical report, https://opendtect.org/osr/Main/NetherlandsOffshoreF3BlockComplete4GB, 2018.

[65] Hua-Wei Zhou. *Practical Seismic Data Analysis*. Cambridge University Press, New York, 1 edition, March 2014. ISBN 978-0-521-19910-0.

[66] S. Venkataraman, Z. Yang, D. Liu, E. Liang, H. Falaki, X. Meng, R. Xin, A. Ghodsi, M. Franklin, I. Stoica, and M. Zaharia. SparkR: Scaling R programs with spark. In *Proceedings of the ACM SIGMOD International Conference on Management of Data*, volume 26-June-2016, pages 1099–1104, 2016.