Eur. Phys. J. Spec. Top. (2026) 235:9119–9137
https://doi.org/10.1140/epjs/s11734-026-02334-7

THE EUROPEAN PHYSICAL JOURNAL SPECIAL TOPICS

Regular Article

Combinatorial time-loops: probabilistic inference on time-series based on recurrence analysis with symbolic methods

Felipe C. Argolo[1,a], Filipe C. Ribeiro[2], Nils Antary[3], Robert G. S. de Araújo[2], Thierry C. Petit Lobão[4], Roberto F. S. Andrade[2], Suani T. R. de Pinho[2], and Norbert Marwan[3,5]

[1] Instituto de Psiquiatria, Universidade de São Paulo, Dr. Ovídio Pires de Campos, 785-Cerqueira César, São Paulo, SP 05403-903, Brazil
[2] Instituto de Física, Universidade Federal da Bahia, Campus Universitário de Ondina-Ondina, Salvador, BA 40210-340, Brazil
[3] Potsdam Institute for Climate Impact Research, Member of the Leibniz Association, P.O. Box 601203, 14412 Potsdam, Germany
[4] Instituto de Matemática e Estatística, Universidade Federal da Bahia, Campus Universitário de Ondina-Ondina, Salvador, BA 40170-110, Brazil
[5] Institute of Physics and Astronomy, University of Potsdam, Karl-Liebknecht-Straße 32, 14476 Potsdam, Germany

Received 11 November 2025 / Accepted 20 April 2026 / Published online 11 May 2026
© The Author(s) 2026

Abstract Recurrence quantification analysis (RQA) is inspired by Poincaré's early studies and describes non-linear characteristics of dynamical systems. It achieves this by identifying similarities between states, pairing each observation with every other. The identification of recurrences maps trajectories to the realm of binary states, which is a fertile ground for combinatorial methods. In this work, we utilize symbolic methods from analytic combinatorics to perform inference on the dynamics of systems represented as time-series data. With combinatorial constructions tailored for special cases, our method provides exact probabilities for inference on small datasets. We demonstrate the detection of significant motifs: specific sequences of consecutive states that are repeated either within a series or between two series. The framework successfully identifies patterns in noisy periodic signals, auto-regressive processes and non-linear systems with chaotic behavior. The methods were implemented and made available in open-source software: `AnalyticComb.jl` and `SymbolicInference.jl`.

1 Introduction

Analyzing time-series patterns is a critical endeavor to investigate the state evolution of dynamical systems with far-reaching implications in diverse fields, such as life sciences, economics, biology, Earth sciences, and finance [2, 3, 23, 33]. These patterns, embedded within sequences of data points, often hold the key to understand underlying processes, to make predictions, and aid decision-making. Time-series data from real world phenomena, characterized by its temporal structure, frequently poses complex analytical challenges, demanding innovative methods to unlock its hidden insights.

Recurrence quantification analysis (RQA) was created for non-linear analysis of dynamical systems by looking at recurrences of their states reconstructed from measured time-series [20–22]. In this context, a recurrence occurs whenever two states at different time instants are either exactly identical or close enough according to a distance threshold. Therefore, recurrence matrices are binary, with entries from the Boolean domain ($B = 0, 1$). The pairwise relationships between all state vectors, in particular specific temporal sequences of such recurrences (recurrence patterns, line structures in the recurrence plot) contain information about characteristic properties of the system such as entropy, predictability, Lyapunov exponents, and dimension spectrum [7, 20].

Some generalizations were done in RQA; for instance, different recurrence metrics were proposed to characterize chaotic [17] and stochastic systems [6, 12], quantifying aspects such as entropy [5].

[a] e-mail: felipe.c.argolo@gmail.com (corresponding author)

The versatility of RQA has made it a valuable tool across multiple fields, including physics, biology, economics, and environmental science. It has been employed in diverse applications, such as characterizing the heartbeat dynamics in cardiology [4, 32], analyzing EEG signals in neurology [27], identifying periodicity in financial market data [8], and even studying climate variability in atmospheric science [23]. RQA's adaptability and broad applicability highlight its utility in elucidating patterns and dependencies within time-series data.

Several approaches have been proposed to perform statistical inference on recurrence patterns. Solutions generally consider the assumption of a binomial distribution for frequencies of recurrences and non-recurrences [1, 13]. However, such choice might not be suitable for inference on complicated problems. For instance, the following binary pattern follows an equiprobable distribution in its overall counts ($p = q = 0.5$). Nonetheless, the inner structure reveals a sequence of alternating blocks with fixed length, make it very unlikely to be sampled from a random binomial distribution:

11111000001111100000111110000001111100000

This paper introduces symbolic methods and analytic combinatorics as powerful tools for time-series analysis, enabling statistical inferences about different systems. By exploring a combinatorial multiverse of similar trajectories, we study probabilities associated with recurrence patterns. In order to achieve this, we leverage the field of analytic combinatorics [10] to derive exact probability distributions for patterns observed in recurrence matrices.

Symbolic methods allow one to describe general objects using a structured grammar of combinatorial operators (e.g. $SEQ(z)$, 'a sequence of'; $PSET(z)$, 'the powerset of'; or 'cycles of' $CYC(z)$). These can be combined to create complex descriptions that are directly mapped into polynomials. For instance, "a sequence of 1's or 0's, followed by a sequence of 1's, followed a sequence of 0's" describes binary strings like '1101011111100000000'. We can use combinatorial operators (e.g. $SEQ(1+0)SEQ(1)SEQ(0)$) to obtain equations that yield counts in such binary strings. This makes it easy to answer problems that would otherwise be hard to solve.

The ability to directly obtain analytic solutions and asymptotic formulas from the construction proves invaluable for addressing inference problems related to recurrence patterns in RQA, along with their implications on predictability, complexity, and regime transitions.

We demonstrate the utility of the method for the detection of arbitrary motifs. Applications outlined include (1) distinguishing signals (e.g. white noise, sine-wave, random-walks) and (2) identifying the number of cycles in different configurations of the logistic map . The notion of a motif was recently formalized [16, 26] to describe subsequences that contain patterns that are very similar to each other in time-series. Characterization of regular sections is key when studying several phenomena. Detecting non-expected time series motifs—repeated segments in a long time series—is relevant for understanding the dynamics mainly in real-world systems [25]. Recently optimized algorithms provide to generate recurrence plots for massive data sets as well as detection of time series motifs [31].

The current paper is organized as follows: Sect. 2 presents a brief overview of recurrence quantification analysis (RQA) and symbolic methods: Sect. 2.1 introduces the recurrence analysis method for dynamical systems and the RQA perspective. Section 2.2 gives an overview the usage of analytic combinatorics to binary words contained in recurrence matrices, including illustrative cases. It introduces the package `AnalyticComb.jl`, with general utilities in combinatorics. Section 3 'Symbolic methods for binary words and RQA' shows procedures to examine binary words in recurrence matrices. Section 4 describes the algorithm that was used to detect significant motifs in time-series, while Sect. 5 describes the methods and dynamical-systems used for applications. Section 6 shows inference on several time-series data(white-noise, sine-wave, noisy sine-wave, auto-correlated signal, logistic map). The last part (Sect. 7) briefly discusses implications and future directions.

2 A brief overview of RQA and symbolic methods

2.1 Time-series and recurrence quantification analysis (RQA)

Time-series data is ubiquitous, ranging from financial market fluctuations to ecological population dynamics and physiological measurements. The analysis of these data often involves deciphering patterns, which may manifest as periodic oscillations, sudden shifts or recurrent trends. Such patterns provide a window into the underlying processes and phenomena, making time-series analysis an essential tool for prediction and decision-making across various domains. Several challenges surround the analysis of non-linear systems, whose patterns are not easy to grasp. In this context, analyzing the relationship between states in the phase space is a useful approach.

Recurrence quantification analysis (RQA) was developed to quantify non-linear characteristics through the pairwise analysis of similar states among all observations in a trajectory. States can be represented by available state variables or by reconstruction from time series using an embedding approach [15]. Since pairs are labeled as either close (1) or not (0), the resulting recurrence matrix is a binary matrix (made of sequences of 0 s and 1 s)[20].

The base of RQA is the recurrence matrix, which is a mapping of the temporal correlations of a sequence of state vectors $\vec{x}_i$ (e.g., derived from time series) onto a two-dimensional matrix $\mathbf{X}(\epsilon)$ with its elements

$$X_{i,j}(\epsilon) = \begin{cases} 1 & \|\vec{x}_i - \vec{x}_j\| \leq \epsilon, \\ 0 & \|\vec{x}_i - \vec{x}_j\| > \epsilon \end{cases} \quad i, j = 1, \ldots, N. \tag{1}$$

Each element (i, j) of the matrix corresponds to a pair of data points in the time series $(\vec{x}_i, \vec{x}_j)$. The value 1 indicates the recurrence of a state (i.e., the states at time i and time j are closer than a chosen threshold ϵ), and 0 otherwise.

The process of defining a recurrence incorporates the parameter ϵ, which determines the error or uncertainty of defining recurrences of states. By plotting the recurrence matrix (recurrence plot), characteristic patterns indicate different dynamics[20] (Fig. 1). Diagonal line structures represent epochs of similar evolution of the dynamical system at different times and can be used to evaluate deterministic dynamics (Fig. 1, left column); a recurrence plot mainly consisting of single points and no or only very few diagonal lines is an indication for stochastic processes (Fig. 1, 2nd left column).

Thus, the different geometrical patterns in the recurrence plot (e.g. recurrence rate; length of diagonal or vertical patterns; distances between recurrence points) illustrate different dynamical aspects and are used to form measures of complexity to quantitatively evaluate the dynamics [20, 22].

Another way of visualizing the pairwise adjacency between all pairs of states is through a graph [19, 35]. In such type of visualization, states (points of time) are considered as nodes of a network and nearby states (recurrences) are represented by links (Fig. 2). The combinatorial approach examines the likelihood of observing certain permutations given possible configurations of links specified by symbolic operators.

2.2 Analytic combinatorics and symbolic methods

Analytic combinatorics provides a systematic approach to analyze complex problems, which were generally tackled using heuristics. It enables the analytic representation of combinatorial objects.

Generating functions, $G(z)$, are one of the key concepts of this approach. They encode counts of combinatorial structures as formal power series, which can be algebraically examined to extract information about the intricate properties of objects.

A brief history of this field goes back to 1751, when Euler was studying the number of ways in which a given convex polygon could be decomposed into triangles by diagonal line [30], he realized that the progression of numbers in the solution.

$$S = 1, 2, 5, 14, 42, 132, \ldots$$

was directly related to the coefficients of the Taylor series expansion of the polynomial fraction

$$\frac{1 - \sqrt{1 - 4z}}{2z} = 1 + 2z + 5z^2 + 14z^3 + 42z^4 + 132z^5 + \cdots$$

The integers in this series (OEIS A000108) came to be known as Catalan numbers, after mathematician Eugène Catalan, who found them when studying the 'Tower of Hanoi' puzzle. They were first described by Chinese mathematician Minggatu [18].

In this case, the function in the left side of equation ($\frac{1-\sqrt{1-4z}}{2z}$) corresponds to the generating function G of the sequence. G serves as a bridge between combinatorial structures and their analytical properties, making it possible to calculate counts and probabilities associated with specific patterns in observed data [10]. In practical terms, it is an algebraic expression whose series expansion yields counts of combinatorial objects. Later on, this association was formalized by the Redfield-Pólya enumeration theorem, which provides an algebraic specification for the number of orbits yielded by group actions on a set.

The symbolic method considers a set of combinatorial objects formed by elements of one or more distinct types and uses a grammar of operators defined on such elements that may be characterized by their size. By atomic elements we mean, for instance, the symbols 0 and 1 that form the binary words. It is known that simple binary words have size 2^n, where n is the number of letters of the word. Let W_2 be the set of binary words of all sizes. It's ordinary generating function is

$$G(z) = W_2(z) = 1/(1 - 2z)$$

for which the coefficients of series of Taylor are 2^z, as we can see in next section.

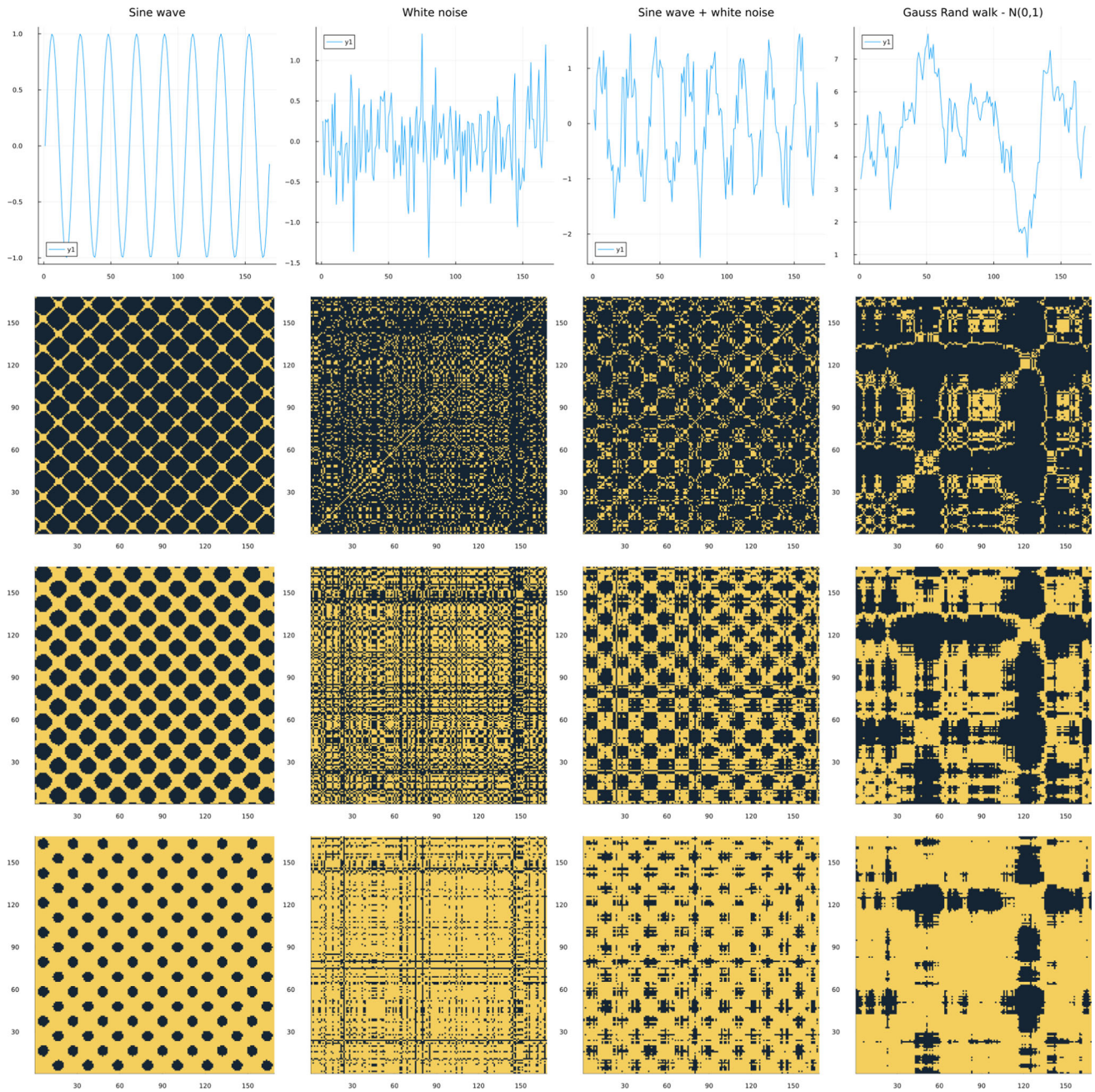


Fig. 1 Original time series (top row) and respective recurrence plots (1st to 4th row) for a periodic signal, white noise, noisy periodic signal, and correlated noise (Gaussian Random Walk, $N(0, 1)$, $a = 0.9$, $c = 0.5$). Each row shows matrices obtained with different recurrence rates (0.2, 0.5, and 0.8)

Some of the basic operators are sum $(+)$, Cartesian product $(\times)$, sequence (SEQ), multiset $(MSET)$, powerset $(PSET)$ and cycle (CYC). Such operators map symbolic constructions onto algebraic expressions $P(z)$, called generating functions, which are directly related to combinatorial objects via complex analysis. [9, 29].

Beside the ordinary generating function associated with the counting of elements, there are generating functions associated with the basic operators referred above. Among those operators, our focus is on the SEQ operator, that is defined by

$$SEQ(\mathscr{W}) = \epsilon + \mathscr{W} + (\mathscr{W} \times \mathscr{W}) + (\mathscr{W} \times \mathscr{W} \times \mathscr{W}) + \cdots$$

where $\mathscr{W}$ a combinatorial class (such as the set of words), $+$ and $\times$ are the sum (disjoint union) and Cartesian product respectively, and ϵ stands for the empty word. For example, if W_1 is a set of unary words (formed by just

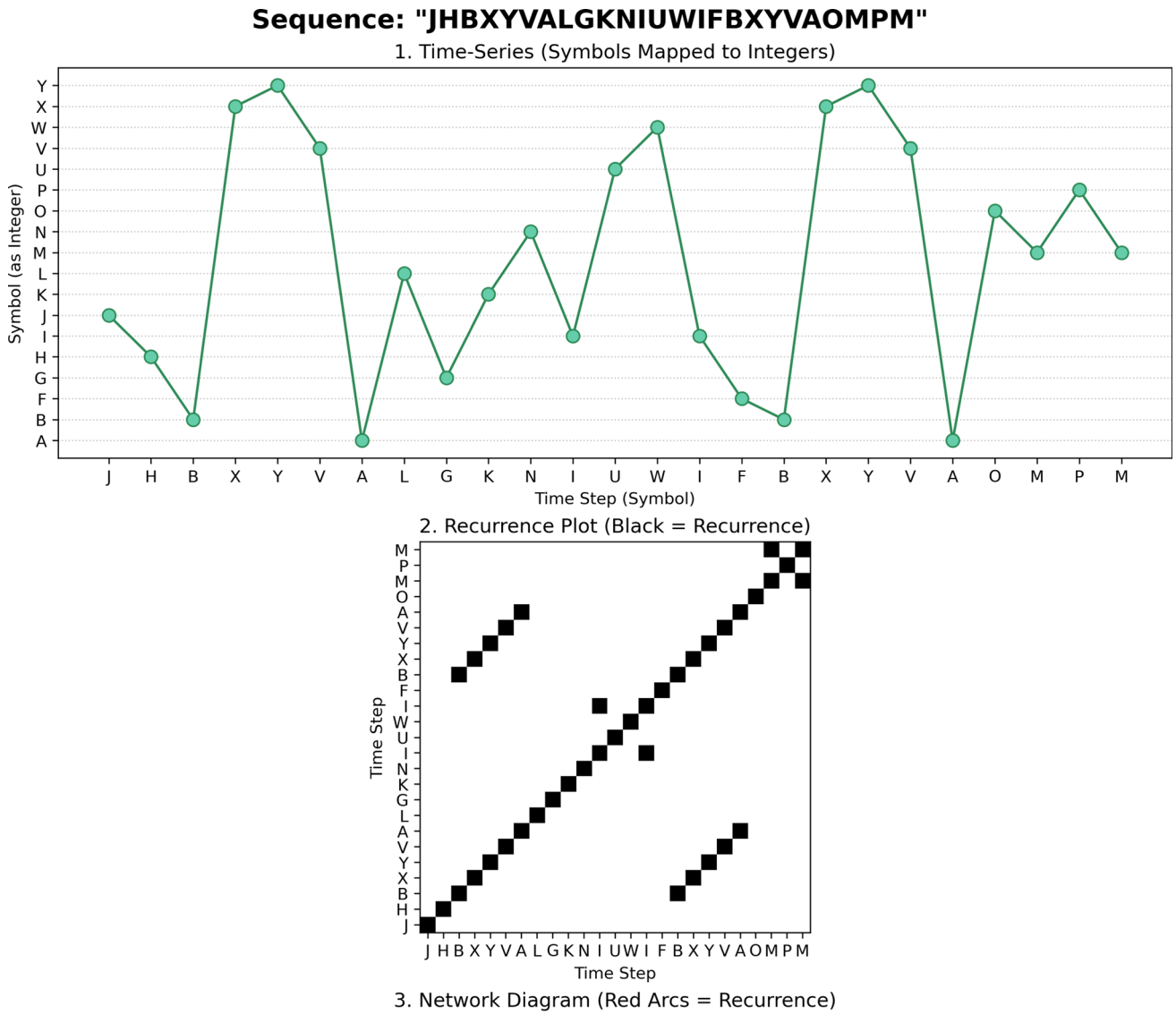


Fig. 2 The top row shows a sequence of symbols, alphabetically mapped to integers. Notice that the 5-letter pattern "BXYVA" is repeated. The middle section displays the appearance of the corresponding long diagonal in the recurrence plot. A network diagram is displayed at the bottom, with red links marking recurrences. In this case, the motif ("BXYVA") is seen at $x_{t=3}$ and once again at $x_{t=17}$

one element), then $SEQ(W_1)$ without ϵ , that may be represented by $SEQ(z)$, is a way of describing the set of positive integers. For the set of binary words, $SEQ(W_2)$ may be represented by

$$SEQ(z+z) = SEQ(2z).$$

Given any contractible combinatorial structure, we can use a set of operators to find a generating function and then approach the problem analytically.

Some examples are exhibited below to illustrate the presented concepts. For binary words, the symbols $\bullet$ or $\star$ with identical sizes will be used to represent elements: $SEQ(\bullet + \star) = SEQ(z+z) = SEQ(2z)$.

Once we have a specification (e.g., $SEQ(z)$, $SEQ(2z)$), obtaining counts of structures in these classes of objects becomes an automatic procedure. We simply need to consider the coefficients of its series expansion. That is, let the series expansion of $G(z)$ be

$$T(G(z)) = \sum_{n=1}^{\infty} a_n z^n$$

then the coefficient values a_n correspond to the counts of objects of size n in this combinatorial class.

As a trivial example, consider firstly the set $\mathscr{W}$ of binary words, which are sequences of elements taken from the binary alphabet $A = \{0, 1\}$,

$$W := \{\epsilon, 0, 1, 00, 01, 10, 11, 000, 001, 010, ..., 1001101, ...\}$$

where ϵ stands for the empty word. The number of binary words of size n is known to be given by $W_n = 2^n$. We will derive this trivial result with the symbolic method to illustrate the standard procedure.

We can use the sequence operator (SEQ) to find the generating function. The sequence operator (SEQ) is also known as Pólya quasi-inverse operator. A sequence of elements (e.g., $\bullet$) of size z is associated with a quasi-inverse map ($SEQ(\bullet) \rightarrow P(z) = \frac{1}{1-z}$). The coefficients in the series expansion of this expression, $T(P(z)) = 1 + z + z^2 + z^3 + \ldots$, are all equal to 1 (where $T(P(z))$ is the Taylor series for $P(z)$, which is the polynomial obtained from the Pólya operator). Therefore, there is exactly one sequence of size 1, ($\bullet$)), one sequence of size 2 ($\bullet\bullet$) , one sequence of size 3 ($\bullet\bullet\bullet$) and so on.

Now, let's refer back to binary words. Binary words are just sequences of distinct symbols, (e.g., $\bullet$ and $\star$), which are tokens of the same size (z).

Then, we have $SEQ(2z)$ as the construction and the series expansion provides the generating function, which encode values of 2^n in the coefficients of z^n,

$$\begin{aligned} \mathscr{W} =& SEQ(\bullet + \star) \implies \mathscr{W} = SEQ(z+z) = SEQ(2z) \\ &\implies W(z) = 1 + (2z) + (2z \times 2z) + (2z \times 2z \times 2z) + \cdots \\ &= 1 + 2z + 4z^2 + 8z^3 + \cdots = T\left(\frac{1}{1-2z}\right) \end{aligned} \tag{2}$$

It contains counts for all possible sequences of both symbols, such as the sequence without symbols (n=1, the empty word); sequences with just one symbol (n=2, $\bullet$ or $\star$), sequences of two (n=4: $\bullet\star$, $\star\bullet$, $\bullet\bullet$, $\star\star$), three (n=8: $\bullet\bullet\bullet$, $\bullet\bullet\star$, $\bullet\star\star$, $\star\star\star$, $\star\star\bullet$, $\star\bullet\bullet$, $\bullet\star\bullet$, $\star\bullet\star$), four (n=16: $\bullet\star\bullet\star$, ...) or five tokens (n=32: $\star\star\star\star\bullet$, ...).

A similar rationale can be used to construct specifications for several combinatorial objects, including constraints on the binary words described above.

3 Symbolic methods for binary words and RQA

The binary sequences of recurrence matrices can be treated as symbolic strings, drawing parallels to concepts from formal language theory and combinatorics. Symbolic methods in this context allow for the calculation of counts, probabilities, and constraints associated with specific patterns within the binary sequences. The following anecdote describes an inference problem analogous to the task at hand and might be used to introduce the subject of inference over binary words: Révész tells the following amusing story attributed to Varga [28]:

> "A class of high school children is divided into two sections. In one of the sections, each child is given a coin which he throws two hundred times, recording the resulting head and tail sequence on a piece of paper. In the other section, the children do not receive coins, but are told instead that they should try to write down

a 'random' head and tail sequence of length two hundred. Collecting these slips of paper, [a statistician] then tries to subdivide them into their original groups. Most of the time, he succeeds quite well."

The statistician's secret is to determine the probability distribution of the maximum length of runs of consecutive letters in a random binary word of length n (here $n = 200$). Students might be able to mimic some of the random features of coin tosses, such as the overall proportion (approximately 1:1, or 100 heads to 100 tails), however, the sequences may present internal structures that are very unlikely to appear at random. For instance, there may be a section with too many consecutive heads. The task of distinguishing coin tosses from fabricated sequences is akin to the one of identifying non-random recurrence patterns in recurrence matrices. In RQA, the ratio of recurrences and non-recurrences might be within expected values, but patterns in subsequences may hint at specific features of the time-series.

The following results illustrate how binary strings can be examined with symbolic constructions. Constructions for all generating functions can be found in the textbook by Flajolet and Sedgewick [10]. Code containing full procedures to obtain counts and plots with software (Julia packages `AnalyticComb.jl`, `SymbolicInference.jl`) is available in the Appendix A.

3.1 A matrix section (e.g., diagonal or column) that contains exactly k recurrences

In this case, we consider the set $\mathscr{W}$ of binary words that contain exactly k occurrences of the letter 1 [11]. We will use symbols $\bullet$ and $\star$ as symbols to avoid confusion, but notice that recurrence matrix entries are 0 ($\bullet$) and 1 ($\star$).

The construction can be thought of: "a sequence of non-recurrences ($SEQ(\bullet)$)", followed by a the pattern "recurrence $\star$ followed by a sequence of non-recurrences, that is $\bullet$ ($\star SEQ(\bullet)$)", which is repeated k times, yielding the specification $((\star SEQ(\bullet))^k)$. This includes sequences such as $\bullet\bullet\star\bullet\bullet\bullet\star\bullet\bullet\bullet\star\bullet\bullet\bullet$ ($k = 3$), $\bullet\star\bullet\bullet\bullet\bullet\star\bullet\bullet\bullet\bullet\star\bullet\bullet\bullet\star$ ($k = 4$) and similar ones (in the recurrence perspective using symbols 1 and 0: 00100010001000, 010000100001000, ...).

The construction and corresponding generating function are given by:

$$\begin{aligned} \mathscr{W} =& SEQ(\bullet)(\star SEQ(\bullet))^k \implies \\ W(z) =& \frac{1}{1-z}\left(z\frac{1}{1-z}\right)^k = \frac{z^k}{(1-z)^{k+1}} \end{aligned} \tag{3}$$

Then, we look for the coefficients of the series expansion to obtain the counts. For instance, among binary words with 10 letters, there are **210** words with **4** $\bullet$s; we use the notation $[z^{10}]T(G(z)) = 210$, for $k = 4$:

$$\begin{aligned} W(z|k=4) =& (1+z+z^2+z^3+\ldots)(z(1+z+z^2+z^3+\ldots))^4 \\ =& 1z^4 + 5z^5 + 15z^6 + 35z^7 + 70z^8 \\ &+ 126z^9 + \mathbf{210}z^{\mathbf{10}} + 330z^{11} + O(z^{12}) \end{aligned} \tag{4}$$

In an evenly distributed recurrence matrix (i.e., $RR = \frac{1}{N^2}\sum_{i,j} R_{i,j} = 0.5$), a section with length 10 will contain exactly 4 recurrences (in any order) with probability 20% ($210/2^{10}$). The counts, which are displayed in the coefficient of the series expansion, actually satisfy $W_n = \binom{n}{k}$, which is a somewhat trivial result. We will advance towards more complex and relevant applications.

3.2 A matrix section (e.g., diagonal or column) that contains exactly k recurrences constrained by a maximum distance d among them

The set $\mathscr{W}$ of binary words that contain exactly k occurrences of the letter 1, constrained by the maximum distance d among occurrences: We can start to harness the power of the symbolic method if we consider constraints on the example above. For instance, we might set a maximum distance between symbol $\star$ (corresponding to the 1 in the recurrence matrix).

The construction is similar to the one above, but using the $SEQ_{<d}$ (sequence with size less than d) operator.

$$\mathscr{W}^{[d]} = SEQ(\bullet)(\star SEQ_{<d}(\bullet))^{k-1}(\star SEQ(\bullet)) \tag{5}$$

If we consider $d = 2$, the maximum distance between $\star$s is 2. Sequences such as 001010101 are allowed, but not sequences like 0100010011.

Using computational tools (see Appendix A), we can quickly find that among binary words with 10 letters and restricted distances, there are 45 words with exactly 4 •s. That is, the coefficient for z^{10} in the Taylor expansion is 45, or $[z^{10}]T(G(z)) = 45$, for $k = 4$.

In an evenly distributed recurrence matrix, a random line with length 10 will contain exactly 4 recurrences spaced by at most 2 non-recurring states with probability 4.4% ($45/2^{10}$). The example with constrained distances is interesting for its complexity, but also seems to bear no relationship with real questions posed at RQA. The next two cases introduce the procedure that will be used to make inference on time-series data.

3.3 Longest runs of recurrences

The length of internal structures in diagonals and columns of recurrence matrices are valuable when making inference about dynamical systems. This is a constrained version of the "coin tosses problem" presented before. However, instead of counting both types of long runs (e.g., consecutive heads or tails), we analyze just one of the outcomes (e.g., consecutive heads).

Considering our binary alphabet (•, ⋆) from above, we want maximum counts for a given letter (e.g., •). The construction is given by restriction on one of the tokens, $\bullet^{<k}$ (alternative notation for $SEQ_{<k}(\bullet)$, or the sum $(z^{k-1} + z^{k-2} + z^{k-3} + \cdots)$, which corresponds to $(1 - z^k)/(1 - z)$. The generating function follows from •s repeated at most k times (including 0), followed by a sequence of ⋆ followed by •s repeated at most k times.

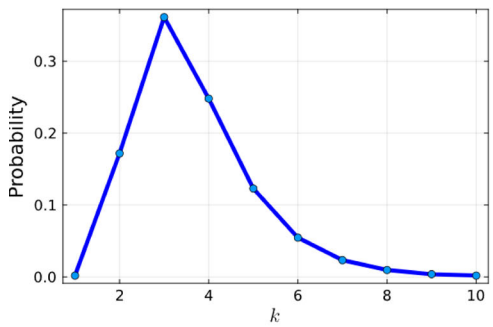
$$\begin{aligned} \mathscr{W}^{<k>} &= \bullet^{<k} \, SEQ(\bullet\star^{<k}) \implies \\ W^{<k>}(z) &= \frac{1 - z^k}{1 - z} \frac{1}{1 - z\frac{1-z^k}{1-z}} = \frac{1 - z^k}{1 - 2z + z^{k+1}} \end{aligned} \tag{6}$$

$W^{<k>}(z)$ is the Ordinary Generating Function (OGF) of words without k consecutive occurrences (e.g. diagonal line of size k). Out of the 1024 10-letter words (2^{10}), only 3 words have sequences of 9 letters or longer: •••••••••• (10 •s), ••••••••••⋆(9 •s:), and ⋆••••••••• (9 •s:). This leaves 1021 words that have sequences of •s with length smaller than 9. Again, one can verify this result in a single step using computational tools.

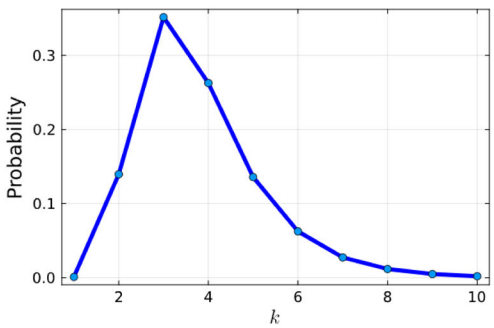
We can calculate probabilities for all possible values of k in order to compute the exact probability distributions (Fig. 3). Once we know the full distributions, it is trivial to perform statistical test for the length of recurrence sequences. Here, the null hypothesis H_0 is that the sequence (in recurrence matrices, diagonal lines) could appear at random.

The plot in Fig. 3a corresponds to the probability distribution ($P(k)$) of random binary words of size 10 without k runs (e.g. diagonal line of size k). Let's consider runs with length 7 (e.g., ⋆⋆•••••••⋆). Sequences with 7-letter runs or longer are among the extreme scenarios associated with this observation.

The one-tailed p-value for observing 7 consecutive recurrences (similar sequence of states of length 7 that appears twice in the time-series) in a diagonal of length 10 is $p = \sum_{k=7}^{k=10} p_k \approx 0.0459$ (see Appendix A for computation and plotting of distributions).



(a) Words of size 10 without k runs.



(b) Words of size 10 with longest consecutive sequence of size k

Fig. 3 Exact probability distributions for two patterns in binary words

3.4 Longest runs of either recurrences or non-recurrences—"the coin problem'

A similar rationale can be used to solve the problem described by the T. Vargas anecdote. Now, we consider both outcomes, counting long runs of either heads or tails at the same time. It is based on the following construction, which describes k-length double runs (either $\bullet$s or $\star$s) in a sequence,

$$\mathscr{W} = SEQ(\star)SEQ(\bullet SEQ(\bullet) \star SEQ(\star))SEQ(\bullet) \tag{7}$$

The section starts with a sequence of recurrences $SEQ(\star)$ and ends with a sequence of non-recurrences $SEQ(\bullet)$. In between, there's a sequence of "non-recurrence followed by a sequence of non-recurrences, followed by a recurrence,followed by a sequence of recurrences $SEQ(\bullet SEQ(\bullet) \star SEQ(\star))$. Notice that $\bullet SEQ(\bullet)$ is equivalent to $SEQ_{>0}(\bullet)$.

We add constraints to sequence sizes (e.g., $SEQ_{<k_1}(\bullet)$, $SEQ_{<k_2}(\star)$). If k_1 and k_2 are equal ($k_1 = k_2 = r$), then the OGF is:

$$\begin{aligned} W^{<r,\,r>}(z) = & \frac{1}{1-2z''} \text{ where } z'' = \frac{z'}{1+z'} \text{ with } z' = \frac{z(1-z^r)}{(1-z)}, \\ & \text{which simplifies to } \frac{1-z^{r+1}}{1-2z+z^{r+1}} = \frac{1+z+\cdots+z^r}{1-z-\cdots-z^r} \end{aligned} \tag{8}$$

The scenario is similar to the previous problem, however both sequences of $\bullet$s and $\star$s are considered.

We can repeat the probabilistic inference test presented previously (H_0: such sequence of consecutive observations could appear at random) for a sequence of length 7 under the new assumption of double-runs. This time, long recurrences (1111...) and non-reccurrences (00000...) are considered, and the one-tailed p-value is $p = \sum_{k=7}^{k=10} p_k \approx 0.0390625$ (Appendix A).

3.5 Longest sequence for arbitrary recurrence rates

Weighted word models—Most of the time, recurrence rates follow unbalanced proportions instead of being evenly distributed (i.e., $RR \neq 0.5$). We can make use of weighted letters to model these cases. Assuming that the probabilities associated with outcomes $\bullet$s and $\star$s are p and q, the probability of the absence of l-length runs in a sequence of n trials is associated with $SEQ_{<\ell}(\bullet)SEQ(*SEQ_{<\ell}(\bullet))$ such that the following multivariate generating function (MGF):

$$W(z,\,u,\,v) = \frac{1-z^\ell u^\ell}{1-zu}\left(1 - zv\frac{1-z^\ell u^\ell}{1-zu}\right)^{-1}$$

with u marking $\bullet$s and v marking $\star$s. After substitutions $u \to p$ an $v \to q$ and taking into account its coefficients $[z^n]$ of powers, it leads to

$$P_{W(z)} = [z^n]\frac{1-p^\ell z^\ell}{1-z+qp^\ell z^{\ell+1}}. \tag{9}$$

4 Symbolic inference: statistical inference on time-series with RQA and symbolic methods

The application of symbolic methods to count elements in recurrence matrix entries is straightforward, as described in the previous section. It illustrates some combinatorial constructions to specify patterns in binary words and how to use them to test statistical hypotheses.

This framework provides infinite possibilities of describing patterns. It can be tailored specifically to a given situation according to the problem at hand.

The Julia package `SymbolicInference.jl` was implemented for that. It utilizes symbolic methods for RQA. It aims to provide a user-friendly, efficient, and scalable solution for researchers and analysts who need to evaluate the RQA results statistically. The Appendix A contains a short tutorial with basic applications.

Algorithm for detection of significant motifs

The following algorithm iterates over each time-lag and considers long-runs of recurrences in them. In a recurrence matrix, this corresponds to off-diagonals, the diagonals that are parallel to the main matrix diagonal. The null

hypothesis assumes a random binary sequence with proportions of recurrences identical to the recurrence rate. The largest sequence in the diagonal is considered and we evaluate the probability of observing an identical or l(identity) diagonal larger sequence at random.

1. Let the ordered set of states in a trajectory S_t be a set of elements s_t labeled from 1 to t.
2. Given a distance function $(d(x, y))$ and a threshold (ϵ), map S_t to a matrix $\mathbf{X}_{t\times t}$ with pairwise adjacency between numbers $(x_{t_i t_j})$ encoded as 1 or 0 $(X = \{x_{t_i t_j} |$ ifelse $(d(s_i, s_j) < \epsilon, 1, 0)\})$. The relative proportions of 1 s and 0 s are p and q.
3. Let M be the set with off-diagonals m_{t_τ} of $X_{t\times t}$: binary strings which contain adjacencies for a given time-lag (m_τ contains the string with adjacencies for $x_{t,\,t+\tau}$: m_1 is the off-diagonal with time-lag of 1, all states $x_{t,\,t+1}$. Conversely, m_2 corresponds to time-lag 2, all states $x_{t,\,t+2}$, and so on).
4. For each binary string $m_{t'<t}$, let K be a list of consecutive runs k_i defined by their starting position (x_t) and size $(card(k_i))$.
5. For each run k_i, the probabilities $P(k—l,\ s,\ p,\ q)$ can be calculated (according to Eq. (9)) given run-size $(l = card(k_i))$, string total size $(s = card(m_i))$ and overall proportion of 1 s and 0 s in the matrix $X_{t\times t}(p,\ q)$.

Where $card(k_i)$ represents the cardinality of k_i, or the number of recurrences in the run.

Instead of working with autocorrelations in a trajectory, one can work with associations between different trajectories $(S1_t,\ S2_t)$ by generating a cross (or joint) recurrence matrix with data from both sets at steps 1 and 2.

5 Illustration of the approach

Our example characterizes several time-series patterns, using recurrent motifs with short and long time-lags. We simulate data from a periodic signal, Gaussian white noise, and auto-correlated noise:

1. Periodic signal (sine wave)
2. Gaussian white noise $\sim N(0,\ 1)$
3. Noisy periodic signal (white noise added to sine wave)
4. Correlated noise ($x_t = ax_{t-1} + c\xi$, with Gaussian white noise $\xi \sim N(0,\ 1)$)
   (a) $a = 0.5; c = 0.5$ (low autocorrelation)
   (b) $a = 0.9; c = 0.5$ (high autocorrelation)

In data pre-processing, all time-series were normalized to have unit energy $(E = \int |f(t)|^2 \mathrm{d}t = 1)$, matching scales between observations.

We consider time-series lengths of 51, 101, and 202.

The p-values were calculated from off-diagonals 1 to 30. Random data is generated with different seed values. Statistics (mean, median, variance, skewness) are calculated with samples of size 300 and 1000.

5.1 Examining the logistic map

In order to address the detection of motifs and patterns in more complex systems, we use time series generated by the standard logistic map

$$x_{n+1} = r \cdot x_n(1 - x_n) \tag{10}$$

in which the value of the control parameter r generates many different types of dynamical behavior, ranging from asymptotic approach to constant behavior, sequence of period doubling cycles, chaotic behavior, and so on. For the purpose of illustrating how the different dynamical behaviors are related to motifs and how they can be detected by our methodology, it is sufficient to consider four different values of r , namely $r = 3.2$, 3.5, 3.56 and 4. These values give rise to trajectories that, respectively, converge to period-2, period-4 and period-8 cycles, and to a chaotic trajectory that emerge after the infinite section of period doubling bifurcations $(r = 4)$.

5.2 Hypotheses and statistical inference

Long-range correlations: We expect harmonic signals to present motifs that occur in intervals agreeing with their fundamental periods. Common patterns should be repeated multiple times, at every interval t. For instance,

this value would be $t = 2\pi$ for classic sine wave. A period of exactly 2π was used, with half-unit (0.5) steps, yielding $\frac{2\pi}{0.5} \sim 12.6$ samples per period. That is, the pattern is repeated approximately at every 12–13 observations.

Using integer number of cycles (4, 8, 16) we obtain time-series with n = 51,101 and 202.

White-noise: We expect pure white-noise time-series to present only high probabilities for all time-lags. That is, no significant motif should be detected in the data.

Short-range correlations: For Auto-regression on x_{t-1}(AR(1)), we expect to find similar states at short time-lags. That is, an arbitrary state is likely to be close to the next state, which is likely to be close to the next one and so on. When analyzing autocorrelation functions (ACF) based on raw time-series, there's a decreasing pattern. Hence, our procedure should detect "abnormal" sequences (diagonal lines of recurrences) in short time-lags. In order to quantify this effect, we adjust linear regression models, considering each off-diagonal and its p-values. Our regression models relate p-values to time-lags. We expect positive slopes for auto-regressive models: less significant correlations for longer time-lags, mirroring an ACF drifting towards zero. This measure is similar to the TREND of a recurrence matrix (e.g. number of recurrent points depending on the distance to the main diagonal) but considering only long sequences of recurrences instead of the overall count per diagonal.

Logistic map: Each configuration of the logistic map (r = 3.2, 3.5, 3.56 and 4) repeated patterns (motifs) at different time-lags, which are expected to be detected with probabilistic inference.

Significance threshold adjustment: In order to avoid false positives, we used Bonferroni correction, in which the p-value threshold is divided by the number of tests being performed. We considered p-values below the threshold of 0.00167 to be significant, since 30 tests for different time-lags ($(x_{t,t+1}, x_{t,t+2}, \ldots, x_{t,t+30})$ were conducted.

5.3 Recurrence rates

We performed a grid-search over different recurrence-rates (0.2,0.4,0.6,0.8) to identify which value could capture significant p-values appearing at different time-lags. This is akin to the study of persistent homologies in the data under different configurations, which has been applied to distance and recurrence matrices [14, 34].

The method `persistence_barcode`, available in `SymbolicInference.jl` uses symbolic inference to help with the exploration of connected components (e.g. significant p-values that persist across neighboring time-lags during grid-search). After probing each series, the value 0.2 was chosen.

6 Results

Distinguishing white noise from periodic and autocorrelated signals: Only white-noise presented large p-values for all off-diagonals. Average and median p-values were above a 0.8 threshold in all scenarios for this class (Fig. 4, 2nd row)

Short-range correlations: Results show an increasing trend in p-values as time spans become longer for auto-regressive models (Fig. 4, 4th and 5th rows). That is, lower-order off-diagonals ($t+1$, $t+2$, …, $t+k$ for small k) are likely to present significant motifs (sequences of recurrences), which is expected in short-ranged auto-correlated signals.

The linear correlation between time-lag and p-values was only significant for the auto-regressive processes, both at low ($a = 0.5$) and high ($a = 0.9$) autocorrelation. The linear coefficient was generally one order of magnitude larger in high-autocorrelation models ($\beta \sim 0.01$) as compared to low-autocorrelation ones ($\beta \sim 0.001$).

Other systems (e.g. periodic sine wave, white noise, sine wave added with noise), did not show sigficant results in this analysis (Suppl. Table 1)

Motifs, long-range correlations and periodicity: The method detects long-ranged sequences of recurrences in both periodic signals. In Fig. 4, the 1st and 3rd rows display an alternating pattern of high and low p-values. According to the system specification (see Sect. 5.2), cycles are expected to repeat at every 12.6 samples. Accordingly, the pure sine wave (1st row) shows very small p-values at time-lags 13 and 25, while other time-lags present high values. The boxplot for time-lags 13 and 25 are near 0, appearing to be almost overlayed with the x-axis in Fig. 4. The 3rd row shows the same sine wave signal added with noise, also showing p-values distributions with smaller quartiles and medians at the neighboring time-lags of 13 and 25.

Logistic map: Fig. 5 indicates the presence or absence of recurrences in each time-lag (1 to 50), considering a time-series of length 300, setting a recurrence rate of 5%.

After an initial transient phase, trajectories with periodic cycles stabilize at states and alternate between them at regular intervals. First, second and third rows display regular patterns in these trajectories. Right column plots highlight the presence or recurrences (1) for even windows (every 2 states). For the 4-period and 8-period cycle, the presence of recurrences is more sparse, at every 3 or 7 states. The fourth row (chaos) shows a presence of recurrences in every time-lag from $t-1$ to $t-50$, highlighting that the system visits many states. Utilizing symbolic inference, we can verify statistically significant motifs displayed on a regular interval, correctly identifying such patterns in the period-2, period-4 and period-8 trajectories. The chaotic configuration displays recurrences for every time-lag

($p < 0.001$). For a threshold of 5% , the chaotic logistic map ($r = 4.0$) presents significant recurring motifs at all time-lags.

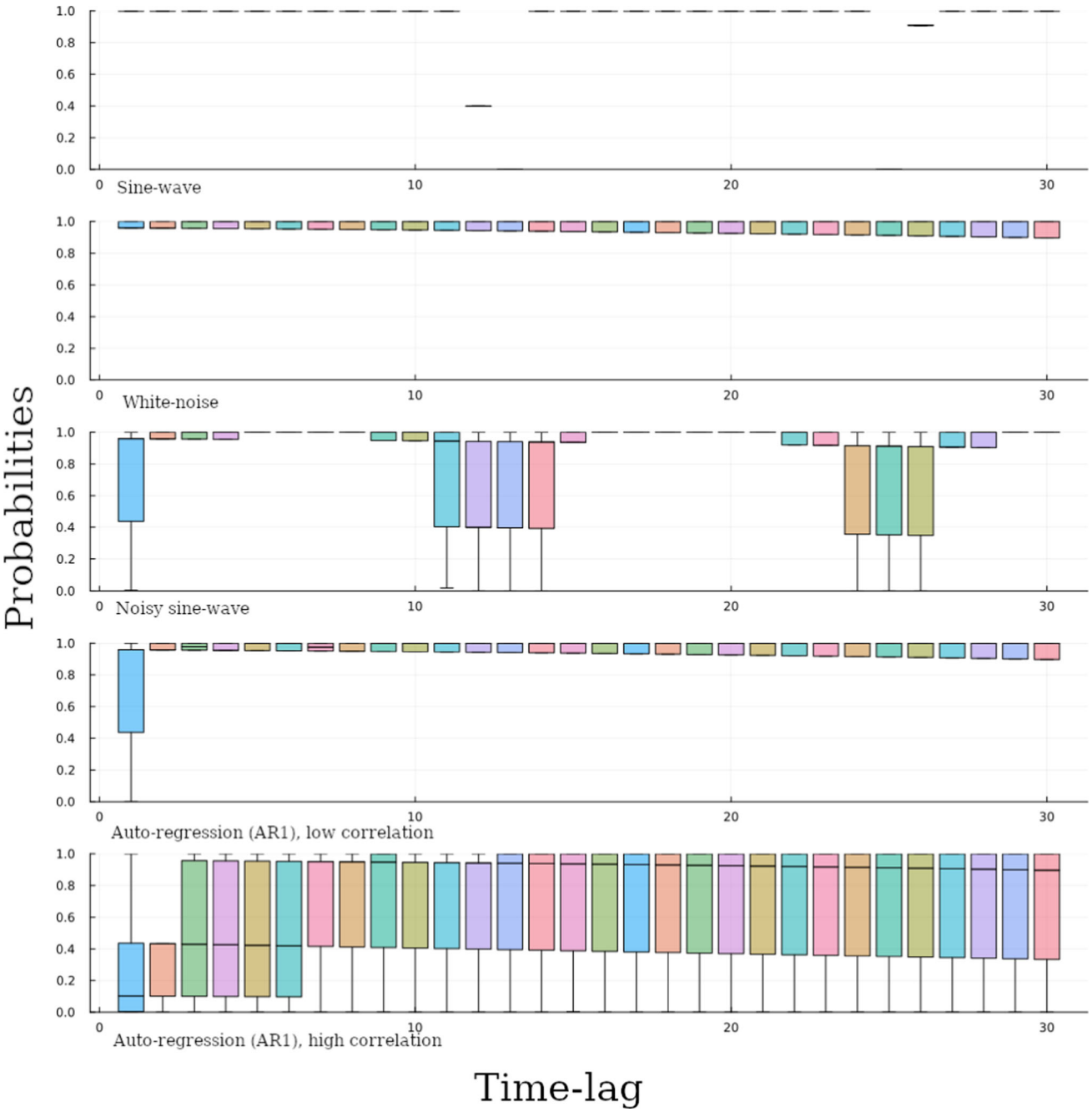


Fig. 4 Boxplot of p-values vs. time-lag examined, considering time-series of size n=101. A sine wave (1st row) contains small p-values in periodic intervals (every 12.6 observations, or 0, $x_{t,\,t+13}$, $x_{t,\,t+25}$). White-noise (2nd) only yields high p-values in all time-lags. A noisy sine-wave (3rd row) reproduces the behavior of a sine-wave (1st row), with more dispersed values and less defined period. Auto regression processes (4th and 5th row) present smaller p-values for short lags (e.g. x_{t+1}) with low autocorrelation, which is more evident in the auto regression with high autocorrelation (5th row)

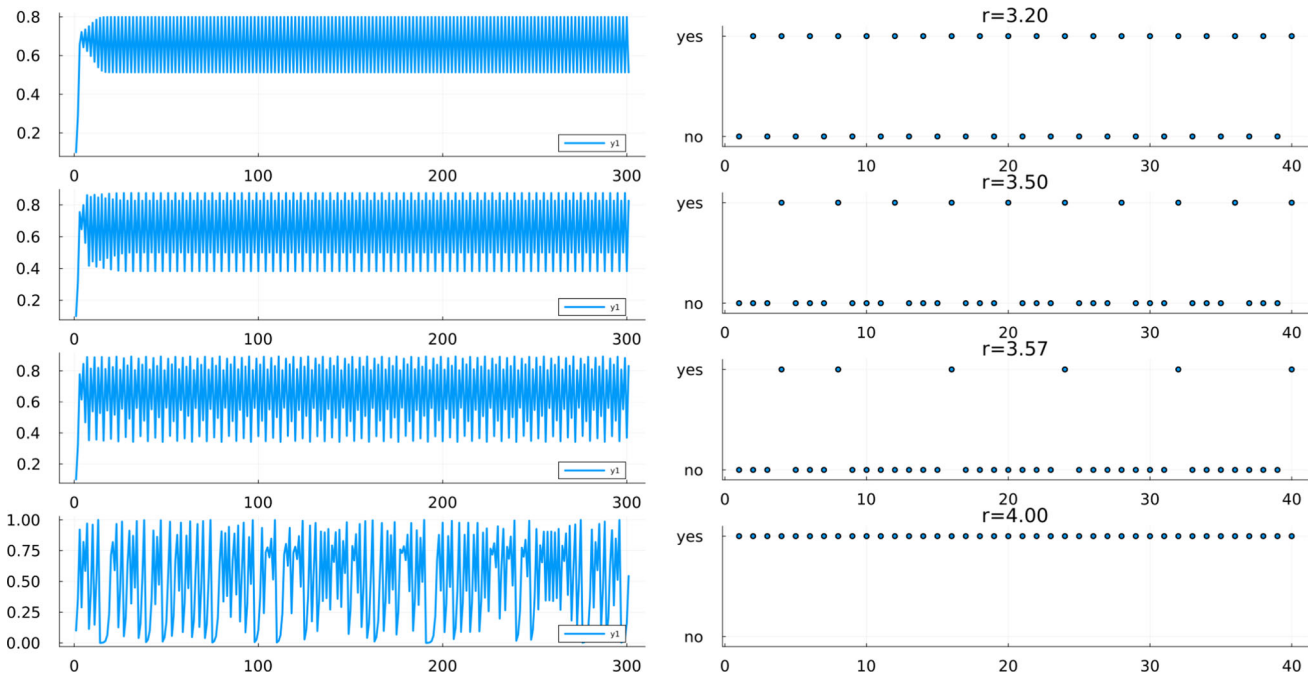


Fig. 5 Left column shows trajectories for different choices of the parameter of r in the logistic map (values on *y-axis* and time on *x-axis*). Right column shows either the absence (0) or presence (1) of recurrences along different time-lags from 1 to 30

7 Discussion

The recurrence quantification analysis (RQA) approach characterizes dynamical systems through their recurrences, mapping structures into the realm of binary words. These objects are a fruitful field for the application of combinatorial tools for inference. We used simulations to show that these methods can successfully distinguish random noise from other series. In addition, symbolic methods may aid in verification of short-range autocorrelations (e.g., those in AR(1)) or patterns occurring over long ranges (e.g., motifs in periodic signals, even when mixed with random noise).

Our methods scan internal structures of recurrence matrices. We detect unusually long sequences of recurrences in diagonals, which are directly associated with the appearance of significant motifs in the original time-series. In addition to the detection of relevant (non-random) off-diagonals, we highlight specific regions of the time-series, which might be invaluable for certain inference problems.

Besides the detection of long runs, symbolic methods allow for specifications of arbitrary patterns using combinatorial operators. For instance, one can look at patterns considering a maximum distance among sequences, or cycles of sequences. Applications of these constructions in RQA context are yet to be explored.

A common obstacle for practical scientists is to make inference when the sample size is small. Statistical methods often consider analytical solutions that assume the existence of large samples to examine asymptotic behavior.

Our methods do not rely on asymptotic behavior of large samples to calculate probabilities. The symbolic method does probabilistic inference with exact counts and probability distributions, being a valuable asset for scenarios with scarce data or short trajectories.

We used a grid-search approach to select the recurrence rate value, $RR = 0.2$. This is a value commonly seen in literature, since it avoids saturation of recurrences. Nonetheless, different choices of recurrence rates lead to different motifs in the recurrence matrix (see (Fig. 1). Therefore, an inadequate choice of RR value may yield very few recurrence (low RR) or too many (high RR, saturated recurrence matrix).

The study of dynamical systems through combinatorial analysis of recurrences is a fairly recent field. Some abstractions that were developed in this venture include 'Morse' and 'brick' decomposition of systems [24]. RQA procedures map a distance matrix into a structure containing information about sufficiently close states, an approach that is also used to identify cycles (a closed path or loop in a topological space) in topological data analysis [14]. Nonetheless, formal results connecting combinatorial constructions of recurrence matrices with symbolic methods and the topology of trajectories remain to be achieved.

8 Supplementary Material

The supplementary material includes full datasets from simulations, including time-series of sizes 51,101 and 202 and replication samples of size 300 and 1000. It also includes a summary table with linear regression coefficients concerning the association of p-values and time-lags in each system. It also includes a figure with boxplot showing the dispersion of such p values in each window for each system.

Supplementary Information The online version contains supplementary material available at https://doi.org/10.1140/epjs/s11734-026-02334-7.

Acknowledgements Kudos to Ricardo Bittencourt for presenting the topic of analytic combinatorics to a wide audience in his personal blog.

Author contribution statement

Felipe—Idea inception, software, analysis, text. Thierry, Norbert, Nils, Suani, Roberto—Discussions, text review—RQA and Dynamical Systems. Filipe Cruz—Software development (contributed to plotting features) and data analysis. Robert—Text organization, discussions.

Funding The Article Processing Charge (APC) for the publication of this research was funded by the Coordenação de Aperfeiçoamento de Pessoal de Nível Superior - Brasil (CAPES) (ROR identifier: 00x0ma614). This work was made possible by FAPESP PRIP (Pro-rectory for Inclusion and Belonging).

Data Availability Julia packages are available directly from official registries. Source can be found in: https://github.com/fargolo/SymbolicInference.jl/; https://github.com/fargolo/AnalyticComb.jl/. Paper examples can be found in https://github.com/fargolo/paper-vignettes/.

Declarations

Conflict of interest Authors declare to have no conflict of interest.

Open Access This article is licensed under a Creative Commons Attribution 4.0 International License, which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence, and indicate if changes were made. The images or other third party material in this article are included in the article's Creative Commons licence, unless indicated otherwise in a credit line to the material. If material is not included in the article's Creative Commons licence and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder. To view a copy of this licence, visit http://creativecommons.org/licenses/by/4.0/.

Appendix A: Code

Two open source Julia packages implement symbolic methods for generic problems (`AnalyticComb.jl`) and their use for inference, including time-series analysis with RQA (`SymbolicInference.jl`). Assuming that Julia is installed in the machine, the following can be typed into the interactive prompt (a.k.a. REPL, read-eval-print loop) to install dependencies and run the examples shown throughout the report.

We start with the class of binary words. The `SEQ` operator is implemented in `AnalyticComb.jl` and can be used to obtain a generating function.

```
# Loads AnalyticComb and SymPy dependencies
julia> using Pkg; Pkg.add(["AnalyticComb", "SymPy"])
julia> using AnalyticComb, SymPy
# Defines z as a symbolic variable
julia> z = SymPy.symbols("z")
# Creates the generating function in W
julia> W = SEQ(2z)
   1
-------
1 - 2z
# Obtains the Taylor series expansion
julia> series(W)
1 + 2z + 4z^2  + 8z^3  + 16z^4  + 32z^5  + O(z)^6
```

Section 3.1: Binary words of length n with exactly k occurrences.
`bin_words_with_k_occurences(k, n)`:

```
julia>bin_words_with_k_occurences(4, 10)
210
```

Section 3.2: Binary words of length 10 with k occurrences constrained by maximum distance d:

```
julia>bin_words_with_k_occurences_constr(4, 10, 2)
45
```

Section 3.3: Binary words of length n without k runs (consecutive occurrences)

```
# parameter 'm' sets the number of symbols. m=2 for binary
    strings.
julia> words_without_k_run(9, 10; m=2)
1021
```

Probability distributions, Fig. 3:
Computation and plotting for exact probability distribution of binary words of size 10 without k-runs.

```
julia>counts = map(x ->
    words_without_k_run(x, 10; m=2), 0:1:10)
julia>probs = diff(counts) ./ 2^10
julia>using Plots; plot(probs)
```

Section 3.4 **- Coin problem**: K-lenght double runs and p-values
Statistical tests in which the null hypothesis H_0 is that the diagonal line (a.k.a. repeated motif in the time-series) could appear at random.
The one-tailed p-value for observing 7 consecutive recurrences (similar sequence of states of length 7 that appears twice in the time-series) in a diagonal of length 10.

```
julia>counts = map(x ->
    words_without_k_run(x, 10; m=2), 0:1:10)
julia>probs = diff(counts) ./ 2^10
julia> sum(probs[7:10])
0.0458984375
```

AnalyticComb.jl— provides a built-in function to calculate probabilities for k-letter sequences in sequences of size n:

```
julia>probs = map(x ->
    bin_words_runs_prob(x, 10), 1:1:10)
10-element Vector{Float64}:
 0.001953125
 0.171875
 0.361328125
 ...
 0.00390625
 0.001953125
julia>plot(probs)
```

We can repeat the probabilistic inference test presented previously (H_0: such sequence of consecutive observations could appear at random) for a sequence of length 7 under the new assumption of double-runs (recurrences and non-reccurrences times are considered).

The one-tailed p-value is now $p \approx 0.0390625$.

```
julia> sum(probs[7:10])
0.0390625
```

Section 3.5: Weighted words

Probabilities associated with consecutive sequences in a slightly skewed distribution of recurrences ($p = 0.4$ and $q = 0.6$) in a section of length 15:

```
julia>raw_probs = map(x->
    weighted_bin_runs_prob(0.4, 0.6, x, 15), 0:1:15)
16-element Vector{Sym}:
 0
 0.000470184984575628
 0.148289149108224
 ...
 0.999998926258176
julia>probs = diff(raw_probs)
15-element Vector{Sym}:
 0.000470184984575628
 0.147818964123648
 0.399568995090432
 ...
 1.04689827841176e-5
 3.22122547191661e-6
julia>plot(probs)

julia> sum(probs[6:15])
0.07098686609817606
julia> sum(probs[7:15])
0.02616500787609599
julia> sum(probs[8:15])
0.009500035645439997
```

Or using a direct call to the built-in function `p_val_weighted`, we can obtain p values with Julia providing recurrence rates (p and q), the number of recurrences (k) and the sequence size (n):

```
julia> weighted_bin_runs_pval(0.4, 0.6, 6, 15)
0.07098686609817606
julia> weighted_bin_runs_pval(0.4, 0.6, 7, 15)
0.02616500787609599
julia> weighted_bin_runs_pval(0.4, 0.6, 8, 15)
0.009500035645439997
```

SymbolicInference.jl—a short tutorial

`SymbolicInference.jl` provides utilities to apply the symbolic method to recurrence matrices in a straightforward way. The following excerpt creates a set of three time-series: sine wave, white noise ($X \sim N(0, 1)$), and the sum of both signals.

```
julia>using Random, Distributions, RecurrenceAnalysis,
   SymbolicInference

julia> rec_rate, sd_norm = 0.6, 0.3
julia> cycles = 12; linear_basis = (0:0.4:2*pi*cycles)
julia> time_series_sin = map(sin, linear_basis)
julia> dist = Distributions.Normal(0, sd_norm)
julia> n_sample = length(linear_basis)
julia>white_n_time_series = rand(dist, n_sample)
julia>time_series_sin_noise = white_n_time_series .+
   time_series_sin
```

-NoValue-Then, let's obtain the recurrence matrix for the noisy sine wave.

```
julia>rec_mat = RecurrenceAnalysis.RecurrenceMatrix(
   time_series_sin_noise, rec_rate; fixedrate=true)
```

We can use `SymbolicInference.jl` utilities to identify significant motifs for the first 15 off-diagonals (e.g., $t+1$, $t+2$, …, $t+15$):

```
julia> motifs_dict = SymbolicInference.rec_matrix_motifs(
   rec_mat; window_range=collect(1:15), seqs="recurrences"
   , n_motifs=2)
 Dict{String, Vector{Any}} with 3 entries:
  "Motifs starts and duration" =>
  [[(16, 7), (18, 6)], ...
  "Probs" => [[0.0838759, 0.258429], ...,
  "Window" => [1, ... , 30]
```

The *motifs_dict* dictionary contains motifs location, their associated p-values and time-lags (`motifs_dict["Motifs starts and duration"]` are tuples of integers indicating starting index and duration of motifs, `motifs_dict["Probs"]` are p-values, and `motifs_dict["Window"]` are the time-lags used).

We can now identify the motif coordinates and plot them:

```
julia> coordinates_dict = extract_recurrences(
    time_series_sin_noise, motifs_dict; num_windows=30)
30-element Vector{Any}:
 Dict{String, Any}
 ("x1" => 56, 57, 58, ...],
 "y1" => [0.866311, 1.53769,...],
 "window" => 29,
 "x2" => [85, 86, 87,  ...],
 "y2" => [-0.499644, 0.352851, ...],
 "size" => 24,
 "prob" => 0.000797918)
 ...
 Dict{String, Any}
 ...

julia> coordinates_dict[1]
# most significant motif
Dict{String, Any} with 7 entries:
  "x1"     => [56, 57, ...]
  "y1"     => [0.866311, 1.53769, ...]
  "window" => 29
  "x2"     => [85, 86, ...]
  "y2"     => [-0.4996, 0.3528,...]
  "size"   => 24
  "prob"   => 0.000797918

julia> p = SymbolicInference.plot_motifs(
    time_series_sin_noise, coordinates_dict; n_motifs=1)
```

References

1. T. Aparicio, E.F. Pozo, D. Saura, An independence test based on joint recurrences. Mod. Econ. **6**(08), 895–907 (2015)
2. F.C. Argolo et al., Time-driven interventions for affective disorders: on resonance and the Oberth effect. Trends Psychiatry Psychother. **42**, 113–114 (2020)
3. P.B. Graben, A. Hutt, Detecting event-related recurrences by symbolic analysis: applications to human language processing. Philos. Trans. R. Soc. A **373**(2034), 20140089 (2014). https://doi.org/10.1098/rsta.2014.0089
4. M. Calderon-Juarez et al., Recurrence quantitative analysis of heart rate variability during intradialytic hypotension. Eur. Phys. J. Spec. Top. **232**, 111–121 (2023). https://doi.org/10.1140/epjs/s11734-022-00688-2
5. G. Corso et al., Quantifying entropy using recurrence matrix microstates. Chaos Interdiscip. J. Nonlinear Sci. **28**(8), 083108 (2018)
6. F.E.L. da Cruz et al., Statistical inference for microstate distribution in recurrence plots. Physica D **459**, 134048 (2024)
7. J.-P. Eckmann, S.O. Kamphorst, D. Ruelle, Recurrence plots of dynamical systems. Europhys. Lett. **4**(9), 973–977 (1987). https://doi.org/10.1209/0295-5075/4/9/004
8. A. Fabretti, M. Ausloos, Recurrence plot and recurrence quantification analysis techniques for detecting a critical regime. Examples from financial market inidices. Int. J. Mod. Phys. C **16**(05), 671–706 (2005)
9. P. Flajolet, Analytic combinatorics: a calculus of discrete structures, in *Proceedings of the Eighteenth Annual ACM-SIAM Symposium on Discrete Algorithms* (2007), pp. 137–148
10. P. Flajolet, R. Sedgewick, *Analytic Combinatorics* (Cambridge University Press, Cambridge, 2009)
11. P. Flajolet, R. Sedgewick, *Analytic Combinatorics. Example 1.9, page 52* (Cambridge University Press, Cambridge, 2009)
12. Y. Hirata, Recurrence plots for characterizing random dynamical systems. Commun. Nonlinear Sci. Numer. Simul. **94**, 105552 (2021)
13. Y. Hirata, K. Aihara, Statistical tests for serial dependence and laminarity on recurrence plots. Int. J. Bifurc. Chaos **21**(04), 1077–1084 (2011)
14. T. Ichinomiya, Time series analysis using persistent homology of distance matrix. Nonlinear Theory Appl. IEICE **14**(2), 79–91 (2023)
15. K.H. Kraemer et al., A unified and automated approach to attractor reconstruction. New J. Phys. **23**, 033017 (2021). https://doi.org/10.1088/1367-2630/abe336

16. J. Lin, E. Keogh, P. Patel, Finding motifs in time series, in *Proc. of the 2nd Workshop on Temporal Data Mining* (2002), pp. 53–68
17. S.R. Lopes et al., Parameter-free quantification of stochastic and chaotic signals. Chaos Solitons Fract. **133**, 109616 (2020)
18. J. Luo, Ming Antu and his power series expansions, in *Seki, Founder of Modern Mathematics in Japan: A Commemoration on His Tercentenary*. (Springer, Tokyo, 2013), pp.299–310
19. N. Marwan et al., Complex network approach for recurrence analysis of time series. Phys. Lett. A **373**(46), 4246–4254 (2009). https://doi.org/10.1016/j.physleta.2009.09.042
20. N. Marwan et al., Recurrence plots for the analysis of complex systems. Phys. Rep. **438**(5–6), 237–329 (2007). https://doi.org/10.1016/j.physrep.2006.11.001
21. N. Marwan, K.H. Kraemer, Trends in recurrence analysis of dynamical systems. Eur. Phys. J. Spec. Top. **232**(1), 5–27 (2023)
22. N. Marwan, C.L. Webber Jr., Mathematical and computational foundations of recurrence quantifications, in *Recurrence Quantification Analysis: Theory and Best Practices*. (Springer, Cham, 2014), pp.3–43
23. N. Marwan et al., Nonlinear time series analysis of palaeoclimate proxy records. Quatern. Sci. Rev. **274**, 107245 (2021)
24. M. Mrozek et al., Combinatorial vs. classical dynamics: recurrence. Commun. Nonlinear Sci. Numer. Simul. **108**, 106226 (2022)
25. A. Mueen, Time series motif discovery: dimensions and applications. Wiley Interdiscip. Rev. Data Min. Knowl. Discov. **4**(2), 152–159 (2014)
26. A. Mueen, et al., Exact discovery of time series motifs, in *Proceedings of the 2009 SIAM International Conference on Data Mining* (SIAM, 2009), pp. 473–484
27. E. Pitsik et al., Motor execution reduces EEG signals complexity: recurrence quantification analysis study. Chaos Interdiscip. J. Nonlinear Sci. **30**(2), 023111 (2020)
28. P. Révész, Strong theorems on coin tossing, in *Proceedings of the International Congress of Mathematicians*, vol. 749 (Helsinki, 1978)
29. M.R. Riedel, Pólya's enumeration theorem and the symbolic method (2006)
30. A. Selim, M. Saravević, Catalan numbers and applications. Int. Sci. J. Vis. **4**(1), 99–114 (2019)
31. M. Shahcheraghi et al., Introducing Mplots: scaling time series recurrence plots to massive datasets. J. Big Data **11**(1), 96 (2024)
32. L. Shinoda et al., Cardiac and autonomic dysfunctions assessed through recurrence quantitative analysis of electrocardiogram signals and an application to the 6-hydroxydopamine Parkinson's disease animal model. Front. Physiol. **12**, 725218 (2021). https://doi.org/10.3389/fphys.2021.725218
33. F. Strozzi, J.-M. Zaldıvar, J.P. Zbilut, Application of nonlinear time series analysis techniques to high-frequency currency exchange data. Physica A **312**(3–4), 520–538 (2002)
34. M. Zhang et al., Topological portraits of multiscale coordination dynamics. J. Neurosci. Methods **339**, 108672 (2020)
35. Y. Zou et al., Complex network approaches to nonlinear time series analysis. Phys. Rep. **787**, 1–97 (2019). https://doi.org/10.1016/j.physrep.2018.10.005