



# An improved estimator of Shannon entropy with applications to systems with memory

Juan De Gregorio, David Sánchez, Raúl Toral\*

*Institute for Cross-Disciplinary Physics and Complex Systems IFISC (UIB-CSIC), Campus Universitat de les Illes Balears, E-07122, Palma de Mallorca, Spain*

## ARTICLE INFO

### Keywords:

Markov processes  
Shannon entropy  
Data analysis  
Systems with memory

## ABSTRACT

We investigate the memory properties of discrete sequences built upon a finite number of states. We find that the block entropy can reliably determine the memory for systems modeled as Markov chains of arbitrary finite order. Further, we provide an entropy estimator that remarkably gives accurate results when correlations are present. To illustrate our findings, we calculate the memory of daily precipitation series at different locations. Our results are in agreement with existing methods being at the same time valid in the undersampled regime and independent of model selection.

## 1. Introduction

Stochastic modeling lies at the heart of many modern studies of complex systems. In this approach, the detailed dynamics is replaced by a set of transition probabilities that connect the states of the system at two different times. Quite often, the resulting stochastic processes can be modeled with the aid of discrete Markov chains [1], where the transition probabilities to a future state depend only on the present state and not on past states [2]. The broad applicability of this Markovian dynamics runs across disciplines: fluctuation theory in statistical physics [3], DNA sequence analysis in molecular biology [4], weather forecast in meteorology simulations [5], or web searches in information retrieval [6], just to mention a few.

However, this memoryless approximation is insufficient if one deals with strongly correlated systems [7,8], whose next state depends not only on the present one but also on a longer history of past states. Higher-order Markov chains [9], which include a finite number  $m$  of past states to determine the transition probability to the next one, are thus needed for an accurate characterization of systems with memory. Recent examples of systems modeled with higher-order Markovian discrete chains include Parkinsonian tremor time series [10], linguistically detected emotional events [11], genomic polymorphisms for cancer research [12], human navigation on the web [13] and autochemotactic searchers [14]. It is therefore of utmost importance to have reliable methods that faithfully determine the order or memory  $m$  for a given sequence of data [15].

This goal can be achieved in several ways. For the purpose of this paper, we focus on information-theoretical methods. The widely used Akaike information criterion (AIC) [16] is based on a loss function constructed from the Kullback–Leibler relative entropy. The loss function is

then minimized to find a good estimate of  $m$  [17]. Since this approach views the transition probabilities from a frequentist perspective, the Bayesian information criterion (BIC) uses instead Bayes factors to further constrain the loss function [18]. While the AIC criterion is shown to overestimate  $m$ , the BIC is biased towards overly simple models [19]. To overcome these difficulties we propose in this paper a new method to determine the order of a Markov chain that best represents a given series of data. Advantages of the proposed method include that it is independent on model selection and sufficiently precise even in the undersampled regime.

Central to our proposal is the block Shannon entropy [20]. Data are grouped in blocks of a given size  $n$  from which the entropy is extracted using the probability of each block. When the total sequence length is much larger than the number of possible blocks, such that all blocks appear with sufficient statistical relevance, the maximum likelihood estimator for the entropy works well. Nevertheless, sequences extracted from real data can be short and the previous estimator is consequently negatively biased [21]. There have been several attempts to derive unbiased estimators for the entropy but, to the best of our knowledge, all these proposals rely on the sequence being composed by independent random states. What is needed is a faithful estimator being able to deal with correlated systems, as inferred in this work.

Therefore, our contribution is twofold. First, we suggest a novel method that determines the memory of a generic discrete sequence when the series is described by a Markov chain of arbitrary order. Second, we introduce an estimator suitable for the expected value of any observable (like the entropy) that provides information of a correlated system. We emphasize that both findings are independent

\* Corresponding author.

E-mail address: [raul@ifisc.uib-csic.es](mailto:raul@ifisc.uib-csic.es) (R. Toral).

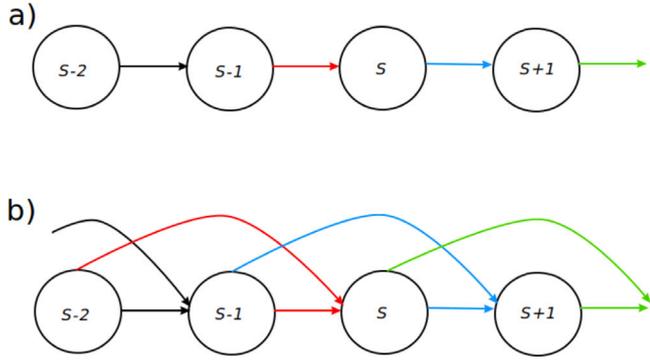


Fig. 1. Sketch for the transition probabilities of two sequences with memory (a)  $m = 1$  and (b)  $m = 2$ .

and constitute, by their own, interesting advances that do not rely on each other. The calculation technique for the order of Markov chains would still hold if another block entropy estimator is used in any of the multiple contexts where these processes are relevant [22]. Additionally, our improved estimator can be applied to any model of correlations (not necessarily a Markov chain) and is thus of interest in situations where entropy is fundamental to understand the system's properties [23].

The paper is organized as follows. In Section 2 we develop the main relation between the block entropy and the order of a Markov chain. We then explain the method to extract the order from the entropy estimator, based on a linear relation between the block entropy and the order, whose proof is left for Appendix. In Section 3 we develop the algorithm to estimate the block entropy from a series of correlated data. In Section 4.1 we present a case study where we determine the memory of a series constructed numerically, while in Section 4.2 we apply the method to determine the order of daily precipitation series, comparing with the results found within the BIC approach. Finally, Section 5 contains our conclusions.

## 2. Relation between block entropy and memory

Consider a discrete random variable  $X$  with  $L$  possible outcomes,  $\{z_i\}, i \in \{1, \dots, L\}$ . Let  $S = (X_1, \dots, X_N)$  be a sequence of time-ordered observations, where  $N$  is the sequence size. Then,  $S$  has memory  $m \geq 1$  if the transition probabilities satisfy

$$P(X_s = z_j | X_{s-1} = z_l, \dots, X_1 = z_k) = P(X_s = z_j | X_{s-1} = z_l, \dots, X_{s-m} = z_r), \quad (1)$$

with  $s$  the time or position in the series. For the case  $m = 0$  the probabilities  $P(X_s = z_j)$  are independent for all  $s$ . The sequence is termed Markovian if  $m = 1$  in Eq. (1). Values  $m > 1$  correspond to higher-order Markov systems, which are the focus of this work. In Fig. 1(a) we illustrate the transition probabilities for a generic sequence with  $m = 1$ : the outcome probability at time  $s$  depends only on the previous state at time  $s - 1$  (red arrow). Fig. 1(b) is a graphical representation for the case  $m = 2$ , where now the probability to observe the system in a given state at time  $s$  depends on states at both previous times  $s - 1$  and  $s - 2$  (red arrows).

In what follows, we consider homogeneous Markov chains for which the  $L^{m+1}$  transition probabilities of Eq. (1) are independent of time index  $s$  and the conditional probabilities are then uniquely determined by the states  $\{z_i\}$ . Furthermore, we will only consider the stationary case and therefore neglect transient effects.

Given a long sequence  $S$  it is not an easy task to determine the order  $m$  by a direct application of Eq. (1) since one should check a growing number of relations as  $m$  is increased. However, a much more efficient procedure stems from an analysis of the block Shannon

entropy [24,25]. Let us form overlapping blocks in  $S$  of size  $n \geq 1$ . The  $j$ th block is then  $B_j^{(n)} = (X_j, \dots, X_{j+n-1})$  with  $1 \leq j \leq N - n + 1 \equiv N_n$ . There exist  $L^n$  distinct possible blocks of size  $n$ , which we denote as  $\{b_i^{(n)}\}_{1 \leq i \leq L^n}$ . All observed blocks  $\{B_j^{(n)}\}_{j=1, \dots, N_n}$  in the series  $S$  belong to the set  $\{b_i^{(n)}\}_{1 \leq i \leq L^n}$ . The block Shannon entropy is defined by

$$H_n = - \sum_{i=1}^{L^n} p(b_i^{(n)}) \log(p(b_i^{(n)})), \quad n \geq 1, \quad (2)$$

where  $p(b_i^{(n)})$  is the probability of appearance of the block sequence  $b_i^{(n)}$ . We use the convention that  $H_0 \equiv 0$ .

The key point of the proposed method to determine the order of a stochastic process is to realize that  $H_n$  is a linear function of  $n$  for  $n \geq m$  if and only if the sequence has memory  $m$  (see Appendix A for a proof). Therefore, we can write

$$H_n = (H_{m+1} - H_m)(n - m) + H_m \quad n \geq m \geq 0, \quad (3)$$

which, for the independent case ( $m = 0$ ), reduces to the known expression for the entropy rate per symbol [26]

$$H_n = nH_1 \quad n \geq 1. \quad (4)$$

For the general case  $m > 0$  the following procedure yields an accurate assessment of  $m$ . Given a trial memory  $\mu = 0, 1, 2, \dots$  we define the trial entropies  $\mathcal{H}_\mu(n)$  by the relation:

$$\mathcal{H}_\mu(n) = (H_{\mu+1} - H_\mu)(n - \mu) + H_\mu \quad n \geq \mu. \quad (5)$$

Now, according to Eq. (3),  $\mathcal{H}_\mu(n)$  would be equal to  $H_n$  for  $n \geq \mu$  if the series  $S$  could be represented by a Markov chain of order  $\mu$ . Therefore, if we consider the mean squared error

$$\Delta_\mu = \frac{1}{n_{\max} - \mu + 1} \sum_{n=\mu}^{n_{\max}} (\mathcal{H}_\mu(n) - H_n)^2, \quad (6)$$

we find that  $\Delta_\mu$  vanishes for  $\mu \geq m$ , independently of the value of the cutoff  $n_{\max}$ . Hence, the criterion to find the order  $m$  of the sequence  $S$  is

$$m = \min(\{\mu : \Delta_\mu = 0\}). \quad (7)$$

In contrast to the BIC and AIC methods, where the loss function depends on the model selected, Eq. (7) is independent of any selection procedure. For memory determination, both AIC and BIC methods are generally used along with a comparison among possible models suitable for a given sequence, whereas our proposed method provides an exact result for discrete random variables. If the system is naturally described with a continuous random variable, we first need to discretize this variable and the results obtained with Eq. (7) will depend on the binsize used for the discretization.

As an example of an explicit calculation of  $m$ , in Fig. 2 we show results for a binary system ( $L = 2$ ) and memory  $m = 3$ . The  $2^{3+1} = 16$  transition probabilities are chosen randomly from a uniform distribution in the interval  $(0, 1)$  (and multiplied by a common factor in order to fulfill the necessary normalization conditions). We plot in the main panel with color lines the trial entropy  $\mathcal{H}_\mu(n)$  given by Eq. (5) for different values of  $\mu$ . For comparison, the exact block entropies  $H_n$ , computed from the transition probabilities, are also shown with black dots. Clearly, the trial entropies  $\mathcal{H}_\mu(n)$  coincide for  $\mu \geq 3$  with the exact block entropy  $H_n$  in agreement with the memory of the process. Alternatively, the inset of Fig. 2 shows that the mean squared error  $\Delta_\mu$  vanishes for  $\mu \geq 3$ . Therefore, Eq. (7) is met and the memory is  $m = 3$ , as expected.

This particular example is based on a particular set of transition probabilities drawn from a uniform distribution. We stress that for a different set of transition probabilities the curves in Fig. 2 would change quantitatively but the condition  $\Delta_\mu = 0$  for  $\mu \geq 3$  will always hold for a system with memory  $m = 3$ . As a consequence, the method is robust against strong variations of the transitions between states.

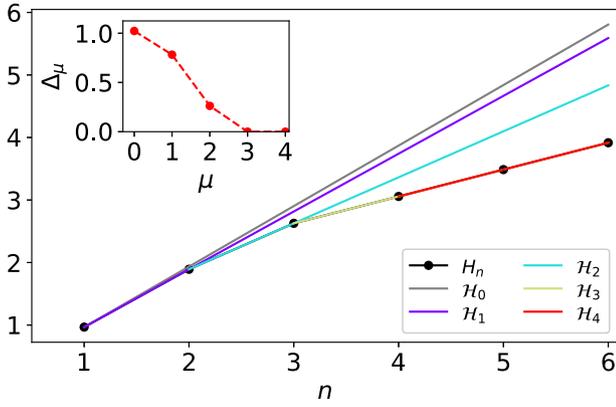


Fig. 2. Block entropies  $H_n$  (dots) and  $H_{\mu(n)}$  ( $\mu = 0, \dots, 4$  with different colors) versus block size  $n$  for a binary sequence,  $L = 2$ , with memory  $m = 3$  and transition probabilities chosen randomly from the interval  $(0, 1)$ . Inset: Mean squared error  $\Delta_\mu$  as a function of the trial memory  $\mu$ . The proposed method determines the true memory from the condition  $\min(\mu : \Delta_\mu = 0)$ , which is satisfied at  $\mu = 3$ .

It should be clear that in this example the determination of the memory  $m$  has been very efficient because we had access to the true block entropies  $H_n$ , determined from the knowledge of the  $L^{m+1}$  transition probabilities. In a case where this is not possible, for example, if we are given a numerical sequence  $S$  of finite length  $N$ , we would need to replace the exact block entropies  $H_n$  in Eqs. (5) and (6) with appropriate estimators  $\hat{H}_n$ . As a consequence, a good performance of our method requires a reliable estimator of the block entropies. Unfortunately, this is not an easy issue since all the entropy estimators reported in the literature have systematic biases that degrade the performance of the method. In the next section, we will propose a new entropy estimator that turns out to be particularly suitable for our purposes of determining the memory of a higher-order Markov process.

### 3. Entropy estimator

Block probabilities and, hence, the Shannon block entropy, could also be obtained with a sufficient precision if we had access to an unlimited amount of data. However, data records are necessarily finite and we must resort to other less accurate estimates of the entropy that take into account this finite amount of data. Hereafter, we will use the notation  $\hat{a}$  to denote a numerical estimator for an arbitrary random variable  $a$ . We recall that an unbiased estimator is one for which  $\langle \hat{a} \rangle = a$ , while for biased estimators the difference  $\langle \hat{a} \rangle - a$  generally decreases with the sample size  $N$ .

A first attempt to estimate the entropy of a general sequence  $S$  is to employ the maximum likelihood estimator (MLE),

$$\hat{H}_n^{\text{MLE}} = - \sum_{i=1}^{L^n} \hat{p}^{\text{MLE}}(b_i^{(n)}) \log \hat{p}^{\text{MLE}}(b_i^{(n)}), \quad (8)$$

where  $\hat{p}^{\text{MLE}}(b_i^{(n)}) = \hat{n}_i^{(n)} / N_n$  is the relative frequency given by the observed number of occurrences  $\hat{n}_i^{(n)}$  for the block  $b_i^{(n)}$  with respect to the total number  $N_n = N - n + 1$  of overlapping blocks of size  $n$  present in  $S$ .

Despite the fact that  $\hat{p}^{\text{MLE}}$  is unbiased,  $\hat{H}_n^{\text{MLE}}$  turns out to be biased [27] and this can lead to an extreme underestimation of  $H_n$ , especially in the undersampled regime  $N_n < L^n$  where  $\langle \hat{H}_n^{\text{MLE}} \rangle < H_n$ . The estimation provided by this method can be considered reliable up to  $n_{\text{max}} \sim \log(N) / \log(L)$ .

There is not known unbiased estimator for the entropy [28] but there is a large number of estimators that manage to improve the results obtained with the MLE [29], allowing one to increase their range of validity as given by  $n_{\text{max}}$ . In particular, the coverage adjusted estimator proposed by Chao and Shen [21] provides good results for sequences of

independent events. Here, we will present an estimator that generally improves the coverage adjusted estimator for systems with memory, using a combination of the Horvitz–Thompson adjustment [30] to account for missing elements in the sequence and a correction to the probabilities that takes into account correlations.

#### 3.1. Horvitz–Thompson correction

Consider a variable  $A(b_i^{(n)})$  that depends on block  $b_i^{(n)}$ . We are interested in a numerical estimation of the sum of  $A$  for all possible blocks, namely  $Y_n = \sum_{i=1}^{L^n} A(b_i^{(n)})$ . In a finite series  $S$ , not all  $L^n$  possible values of  $b_i^{(n)}$  will necessarily appear. To account for the missing blocks in the data, Horvitz and Thompson [30] proposed to estimate  $Y_n$  by summing only the contributions of the blocks that do appear in  $S$  and weighting each term by the probability that the element is included in  $S$ :

$$\hat{Y}_n^{\text{HT}} = \sum_{b_i^{(n)} \in S} \frac{A(b_i^{(n)})}{P(b_i^{(n)} \in S)}, \quad (9)$$

For a random sequence of length  $N$ , memory  $m = 0$  (independent events) and block size  $n = 1$ , the probability of appearance for block  $b_i^{(1)}$  can be computed as

$$P(b_i^{(1)} \in S) = 1 - (1 - p(b_i^{(1)}))^N. \quad (10)$$

In principle, Eq. (10) does not hold in the presence of correlations ( $m > 0$ ). It does not hold either for  $m = 0$  and  $n > 1$  as the existing overlapping between consecutive blocks already induces correlations in the block series: e.g., in the case  $L = 2$  and  $n = 3$  with possible results  $z_i = 0, 1$ , the block  $(0, 0, 1)$  can only be followed by the blocks  $(0, 1, 0)$  and  $(0, 1, 1)$ . Nevertheless, we have checked in all our numerical simulations that the corrections introduced by these effects are negligible in the limit  $N \gg n$  (see Appendix B).

Using Eqs. (9) and (10) in Eq. (2), one arrives at the Horvitz–Thompson estimator for the block entropy

$$\hat{H}_n^{\text{HT}} = - \sum_{b_i^{(n)} \in S} \frac{p(b_i^{(n)}) \log(p(b_i^{(n)}))}{1 - (1 - p(b_i^{(n)}))^N}. \quad (11)$$

#### 3.2. Chao–Shen estimator

In general, the exact probabilities  $p(b_i^{(n)})$  that appear in Eq. (11) are not known. Thus, it is necessary to replace them by their estimators obtained from the sequence  $S$ . The most basic estimator is the maximum-likelihood-estimator  $\hat{p}^{\text{MLE}}(b_i^{(n)})$  defined above. However, since not all possible blocks are likely to appear in the finite sequence  $S$ , we note that  $\sum_{b_i^{(n)} \in S} p(b_i^{(n)}) \leq 1$  but, by construction,  $\sum_{b_i^{(n)} \in S} \hat{p}^{\text{MLE}}(b_i^{(n)}) = 1$ .

Therefore, it is convenient to correct the estimated frequencies before substituting them into Eq. (11). Chao and Shen [21] proposed to use instead of  $\hat{p}^{\text{MLE}}(b_i^{(n)})$  a new estimator  $\hat{p}(b_i^{(n)})$  obtained by multiplying  $\hat{p}^{\text{MLE}}(b_i^{(n)})$  by the *sample coverage*  $\hat{C}_n$ ,

$$\hat{p}(b_i^{(n)}) = \hat{C}_n \hat{p}^{\text{MLE}}(b_i^{(n)}). \quad (12)$$

The new estimator is requested to satisfy the condition  $\sum_{b_i^{(n)} \in S} \hat{p}(b_i^{(n)}) = \sum_{b_i^{(n)} \in S} p(b_i^{(n)})$ . From Eq. (12) and using the normalization condition of the MLE estimator, one finds that

$$\hat{C}_n = \sum_{b_i^{(n)} \in S} p(b_i^{(n)}) \quad (13)$$

represents the total probability for the occurrence of the blocks observed in  $S$ .  $\sum_{b_i^{(n)} \in S} p(b_i^{(n)}) < 1$  implies that there exist unseen blocks in the data and this is an error source for the estimation of  $H_n$ .

Again, as the exact values of  $p(b_i^{(n)})$  are not known, Eq. (13) cannot be used to compute  $\hat{C}_n$  for a given sequence  $S$ . The Good–Turing estimator [31] can be used as an estimate for the sample coverage,

$$\hat{C}_n^{\text{GT}} = 1 - \frac{N_1^{(n)}}{N_n}, \quad (14)$$

where  $N_1^{(n)}$  is the number of blocks of size  $n$  that appear only once in the sequence  $S$ . Substituting Eq. (12) into Eq. (11) and using Eq. (14), one finds the Chao–Shen entropy estimator

$$\hat{H}_n^{\text{CS}} = - \sum_{b_i^{(n)} \in S} \frac{\hat{C}_n^{\text{GT}} \hat{p}^{\text{MLE}}(b_i^{(n)}) \log(\hat{C}_n^{\text{GT}} \hat{p}^{\text{MLE}}(b_i^{(n)}))}{1 - (1 - \hat{C}_n^{\text{GT}} \hat{p}^{\text{MLE}}(b_i^{(n)}))^{N_n}}, \quad (15)$$

which has proven to provide very good results for independent sequences. However, we show below that Eq. (15) does not work so well for correlated data as happens in systems with memory. This is our motivation to present an improved estimator that does take into account correlations.

### 3.3. Correlation coverage estimator

Our proposal follows Eq. (12) but  $\hat{C}_n$  is now estimated using a sequential procedure to tackle possible correlations in the sequence. First, we consider the initial part of the sequence, namely  $S_0 = (X_1, X_2, \dots, X_{N_1})$ , with  $N_1 \equiv n - 1 + N_n/2$ , such that  $S_0$  contains exactly  $N_n/2 \equiv N'_n$  blocks of size  $n$ . We adopt the initial estimator  $\hat{C}_n^{(0)} = 1$ . After observing all the blocks  $B_{j=1, \dots, N'_n}^{(n)}$  that appear in  $S_0$ , we take the next observation  $B_{+1}^{(n)}$  (we adopt the simplifying notation  $B_{+k}^{(n)} \equiv B_{N'_n+k}^{(n)}$ ,  $1 \leq k \leq N'_n$ ), and modify the sample coverage according to

$$\hat{C}_n^{(1)} = \begin{cases} \hat{C}_n^{(0)} & \text{if } B_{+1}^{(n)} \in S_0, \\ \hat{C}_n^{(0)} - \frac{1}{N'_n + 1} & \text{if } B_{+1}^{(n)} \notin S_0, \end{cases} \quad (16)$$

here the factor  $1/(N'_n + 1)$  accounts for the probability that the observed block  $B_{+1}^{(n)}$  appears for the first time. We repeat the process with the next observation  $B_{+2}^{(n)}$  to modify the estimator

$$\hat{C}_n^{(2)} = \begin{cases} \hat{C}_n^{(1)} & \text{if } B_{+2}^{(n)} \in S_{+1}, \\ \hat{C}_n^{(1)} - \frac{1}{N'_n + 2} & \text{if } B_{+2}^{(n)} \notin S_{+1}, \end{cases} \quad (17)$$

where  $S_{+i} = S_0 \cup (X_{N_1+1}, \dots, X_{N_1+i})$ . At variance with the coverage given by Eq. (14), which is obtained from a single observation from the whole series, this procedure has the advantage of iteratively updating  $\hat{C}_n^{(k)}$ ,  $k = 0, 1, 2, \dots, N'_n$  based on the previously observed data. We continue with this procedure until we arrive at the final value of the correlation coverage (CC) estimator

$$\hat{C}_n^{\text{CC}} \equiv \hat{C}_n^{(N'_n)} = 1 - \sum_{j=1}^{N'_n} \frac{1}{N'_n + j} I(B_{+j}^{(n)} \notin S_{+(j-1)}), \quad (18)$$

where the indicator function  $I(Z)$  yields 1 if the event  $Z$  is true and 0 otherwise.

Finally, we substitute the corrected probabilities into Eq. (11) to obtain the correlation coverage-adjusted entropy estimator

$$\hat{H}_n^{\text{CC}} = - \sum_{b_i^{(n)} \in S} \frac{\hat{C}_n^{\text{CC}} \hat{p}^{\text{MLE}}(b_i^{(n)}) \log(\hat{C}_n^{\text{CC}} \hat{p}^{\text{MLE}}(b_i^{(n)}))}{1 - (1 - \hat{C}_n^{\text{CC}} \hat{p}^{\text{MLE}}(b_i^{(n)}))^{N_n}}. \quad (19)$$

As an example, in Fig. 3 we show the exact entropy  $H_n$  (dashed black line and dots) as a function of the block size for a Markovian ( $m = 1$ ) binary process that takes the values  $z = 0, 1$ , with transition probabilities  $P_0 \equiv p(0|0) = P(X_s = 0|X_{s-1} = 0) = 0.7$  and  $P_1 \equiv p(1|1) = P(X_s = 1|X_{s-1} = 1) = 0.6$ . Note that, as expected,  $H_n$  is linear from  $n \geq 1$ . To compare with, we also depict the results obtained

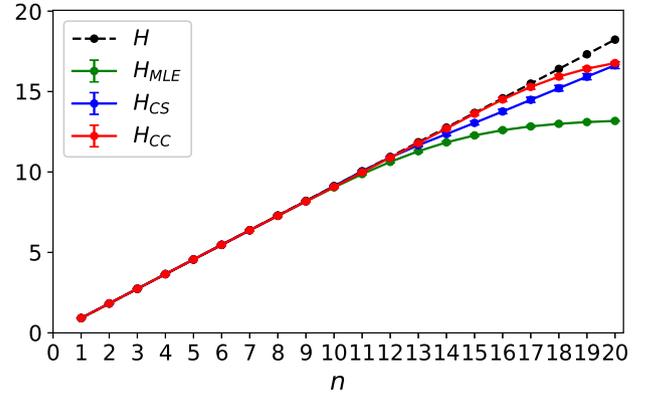


Fig. 3. Exact Shannon entropy per block of size  $n$  (dotted line) for a particular case of a Markovian binary system with fixed transition probabilities  $p(0|0) = 0.7$  and  $p(1|1) = 0.6$ . A sequence of  $N = 10^4$  realizations is numerically generated from which we calculate three different entropy estimators discussed in the main text. The best performance is shown for the estimator that takes into account correlations (red dots).

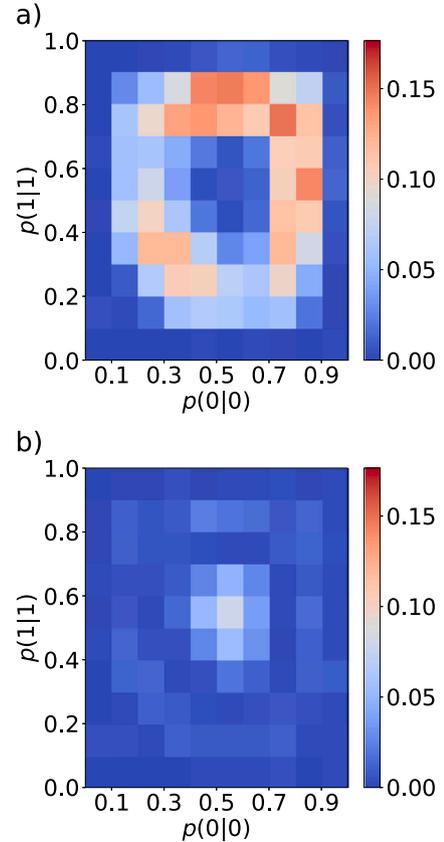


Fig. 4. Comparison between the Chao–Shen entropy, panel (a), and our proposed estimator for correlated systems, panel (b). Calculations are done for a binary system with memory  $m = 1$  and varying transition probabilities  $p(0|0) \equiv P_0$  and  $p(1|1) \equiv P_1$ . Colors represent the departure of each estimator (computed from  $10^4$  realizations) from the exact entropy, as measured by the mean squared error  $\epsilon(P_0, P_1) \equiv \frac{1}{n_{\text{max}}} \sum_{n=1}^{n_{\text{max}}} (H_n - \hat{H}_n)^2$  with  $n_{\text{max}} = 17$ . Adding all values for  $\epsilon(P_0, P_1)$  using the grid size  $\Delta P = 0.1$ , we find that the overall error reaches a value of 4.65 in (a) but only of 0.90 in (b).

with the maximum-likelihood estimators given by Eq. (8) (green line and dots), the Chao–Shen estimator of Eq. (15) (blue line and dots) and our coverage-adjusted estimator proposed in Eq. (19) (red line and dots), all calculated from a sequence of  $N = 10^4$  elements generated numerically. Notably, the proposed estimator performs better than both the MLE and Chao–Shen entropies and perfectly agrees with the exact

result for a wide range of block sizes up to  $n \lesssim 17$  while the MLE and the Chao–Shen estimators deviate earlier from the exact  $H_n$ . In fact, the MLE estimator provides a good approximation up to a value of  $n \lesssim 12$  which is close to the expected limit  $n \sim \log(N)/\log(L) \sim 13$ , while the validity of the Chao–Shen estimator extends up to  $n \lesssim 13$  and our estimator even further. In Appendix C we present a similar comparison for sequences with memory 2 and 3.

Departures from the exact value will show up only in the extremely undersampled regime ( $N_n \ll L^n$ ). In this case, every observation will constitute a different element. As a consequence, the sum in Eq. (19) will include all blocks in the second part and  $\hat{C}_n^{CC} = 1 - \sum_{j=1}^{N'_n} \frac{1}{N'_n+j} \simeq 1 - \ln(2) \simeq 0.307$ , for large  $N$ . The proximity of the computed estimator to this limiting value indicates a largely undersampled sequence and determines the validity of the estimators to the coverage  $C_n$ .

For a more detailed comparison, we show in panel (a) of Fig. 4 the performance of the Chao–Shen estimator given by Eq. (15), and in panel (b) that of our correlation estimator from Eq. (19), both calculated for a process with  $m = 1$  and  $L = 2$ . We show the results as a function of all possible combinations of transition probabilities  $P_0$  and  $P_1$ , the remaining probabilities being derived from the normalization conditions  $p(1|0) = 1 - P_0$  and  $p(0|1) = 1 - P_1$ . We measure the goodness of each estimator for particular values of  $P_0$  and  $P_1$  with the mean squared error  $\epsilon(P_0, P_1) \equiv \frac{1}{n_{\max}} \sum_{n=1}^{n_{\max}} (H_n - \hat{H}_n)^2$ . To improve the statistics of the error, for each set of transition probabilities  $(P_0, P_1)$  we generate  $M = 20$  series, each with  $N = 10^4$  elements, and average the mean squared error over the  $M$  values. We use in all cases  $n_{\max} = 17$  and plot the resulting values in a color code. As shown in the figure, quite generally, Eq. (19) performs better than Eq. (15), with the possible exception of the cases near  $P_0, P_1 \approx 1/2$  (the independent case with equal outcome probability), in which the latter is slightly better. The overall performance is obtained by adding all values of  $\epsilon(P_0, P_1)$  using a grid size  $\Delta P = 0.1$  in Fig. 4, resulting in an aggregated mean squared error five times larger for the Chao–Shen estimator as compared with the correlation coverage-adjusted estimator.

#### 4. Determination of the memory of a sequence

Having demonstrated the usefulness of the estimator given by Eq. (19) for Markovian systems, we now return to the method of Section 2 for the determination of the memory in discrete sequences.

Suppose we are given a finite time series  $S$  that describes a phenomenon for which we would like to determine its memory  $m$ . As explained above, we will use the criterion given by Eq. (7) after computing  $\Delta_\mu$  following Eqs. (5), (6) and using a suitable estimator  $\hat{H}_n$  for the block entropy  $H_n$ . For a given entropy estimator with a known  $n_{\max}$  and fixed  $N$  and  $L$ , a meaningful linear fit for the calculation of  $\Delta_\mu$  as given by Eq. (6) requires that  $\mu \leq n_{\max} - 2$  (at least three points are required for a meaningful fit to a straight line). Since the chosen entropy estimator works for block sizes up to  $n_{\max}$ , our method can provide in principle an accurate result if the system under study has memory  $m \leq n_{\max} - 2$  but would fail otherwise.

In Figs. 3 and C.1 we have shown evidence that  $n_{\max}^{CC} > n_{\max}^{MLE}$ . Hence, there exist three regimes for the estimation of the memory  $m$ : (i) if  $m \leq n_{\max}^{MLE} - 2$ , the fit to a straight line works with both MLE and CC entropy estimators; (ii) if  $n_{\max}^{MLE} - 2 < m \leq n_{\max}^{CC} - 2$ , the MLE estimator is insufficient; (iii) if  $m > n_{\max}^{CC} - 2$  the memory is exceedingly large and both estimators are inadequate.

It is worth mentioning that, in general, the value of  $n_{\max}$  for a certain estimator is not the same for every sequence since  $n_{\max}$  may depend on the transition probabilities. To be conservative, in this section we use  $n_{\max}^{CC} = n_{\max}^{MLE}$  even though our estimator certainly provides good results for larger block sizes.

Due to the limitations of all estimators we expect that  $\Delta_\mu$  is determined within an error that must be taken into account. To do so, we consider  $M$  series of data  $S^{(i)}$ ,  $i = 1, \dots, M$ , all with the same length,

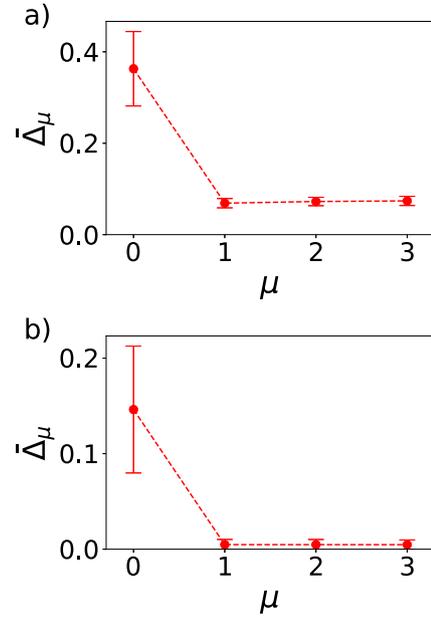


Fig. 5. Mean squared error  $\bar{\Delta}_\mu$  as a function of the trial entropy with  $n_{\max} = 10$  after averaging over  $M = 20$  realizations of a Markov chain with  $N = 1000$  data points, memory  $m = 1$  and transition probabilities  $p(0|0) = 0.7$  and  $p(1|1) = 0.6$ . In panel (a) we use the MLE estimator while panel (b) is generated with the correlation coverage-adjusted estimator. Error bars are given by the standard deviation. Importantly, our proposed estimator clearly yields the memory value  $m = 1$  applying Eq. (20).

obtained either by repeating the experiment  $M$  times or by splitting the original series in  $M$  disconnected sequences. For each sequence we calculate the corresponding value of  $\Delta_\mu^{(i)}$ . Then, we calculate the mean  $\bar{\Delta}_\mu$  and standard deviation  $\sigma_\mu$  for the obtained values  $\Delta_\mu^{(i)}$ . The condition given by Eq. (7) is transformed into the criterion that the mean value  $\bar{\Delta}_\mu$  is consistent with the value 0 within the standard deviation,

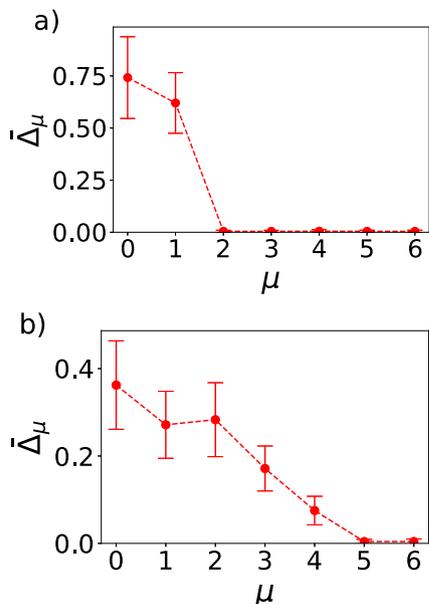
$$m = \min(\mu : \bar{\Delta}_\mu - \sigma_\mu \leq 0). \quad (20)$$

##### 4.1. Numerical simulation

We first present results arising from controlled numerical simulations. We generate  $M = 20$  series of length  $N = 1000$  with memory  $m = 1$  and possible values  $z = 0, 1$  ( $L = 2$ ). We show only results corresponding to  $p(0|0) = 0.7$  and  $p(1|1) = 0.6$  but similar conclusions are obtained quite generally for different values of the transition probabilities. In Fig. 5 we plot the values of  $\bar{\Delta}_\mu$  obtained using the method explained above. In Fig. 5(a) we employ the MLE estimator for the block entropy, while Fig. 5(b) uses the correlation coverage-adjusted estimator. Remarkably, the MLE entropy is not able to yield any useful result, and it is thus not possible to determine the sequence memory since Eq. (20) is never satisfied. In stark contrast, the calculation of  $\bar{\Delta}_\mu$  and  $\sigma_\mu$  using the entropy estimator given by Eq. (19) clearly shows that the criterion Eq. (20) yields the correct result  $m = 1$ .

In Fig. 6(a) and (b) we present results arising from similar numerical simulations as before but now generating sequences with memory  $m = 2$  and  $m = 5$ , respectively (same values of  $M$ ,  $N$  and  $L$  as before) and transition probabilities chosen randomly from an uniform distribution in the interval  $(0, 1)$ . We plot the values of  $\bar{\Delta}_\mu$  obtained using the correlation coverage-adjusted estimator for the entropy. Again, the usage of our proposed entropy estimator allows us to accurately determine the memory of the system. Similar conclusions are generally obtained for a different set of transition probabilities.

These numerical experiments illustrate the importance of having a reliable entropy estimator to successfully apply the method of memory determination.



**Fig. 6.** Mean squared error  $\bar{\Delta}_\mu$  as a function of the trial entropy with  $n_{\max} = 10$  after averaging over  $M = 20$  realizations of a Markov chain with  $N = 1000$  data points, transition probabilities chosen randomly from a uniform distribution in the interval  $(0, 1)$  and memory  $m = 2$  panel (a), and  $m = 5$ , panel (b), using the method explained in the main text with correlation coverage-adjusted estimator. Error bars are given by the standard deviation. Importantly, in both cases, our proposed entropy estimator clearly yields the right memory value applying Eq. (20).

#### 4.2. Daily precipitations

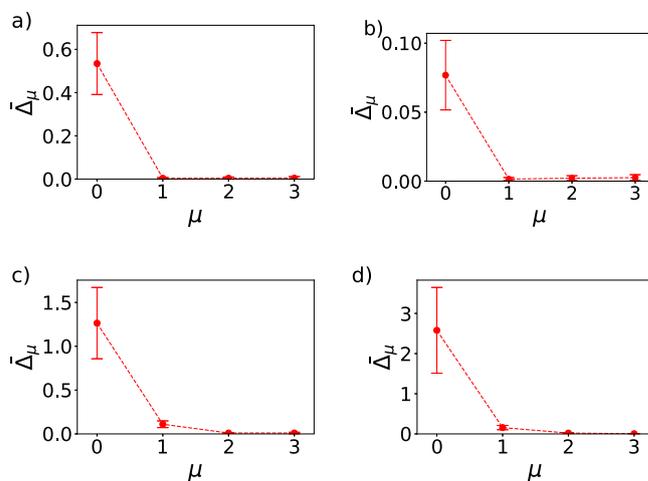
We have thus far tested our proposed method only with numerically generated sequences of known memories. We now test the method with real data. It has been widely accepted that the occurrence (or not) of precipitation can be modeled as a system of memory 1, but it is also well known that this assumption has several shortcomings [32]. There have been some attempts to improve this model by studying sequences of daily data worldwide. It has been found [33] that the model memory depends on the geographical location. Here, we select a few locations and compare the results with those obtained using the BIC method [34].

We collect data from the Global Historical Climatology Network Daily [35]. For each location, we record the available observation for daily precipitation, setting the threshold at 0.1 mm to specify if a day is rainy or not. This way we produce a time series with two states ( $L = 2$ ) and length  $N_s \lesssim 25000$  (the exact value of  $N_s$  varies for each location). Then, this series is divided in  $M = 5$  sequences of equal lengths  $N = N_s/5$  and for each of this sequence we estimate the block entropies  $\hat{H}_n^{CC}$  for  $n = 1, \dots, 12$ , from which we obtain the mean values  $\bar{\Delta}_\mu$  and their standard deviations  $\sigma_\mu$ . It should be noted that the use of the correlation coverage estimator for the entropy allows us to obtain reliable results up to  $n_{\max} = 12$ . If we had used the MLE estimator, this limit value would have been  $n_{\max} \sim 11$ .

In Fig. 7 we show the results obtained for four locations: Rome, Dallas, Bangkok and Than Lwin. For the first two locations, our method predicts  $m = 1$  whereas for the second two places the procedure suggests that both series are better described with  $m = 2$ . These results are in excellent agreement with the BIC method (see Fig. 5 in Ref. [33]) and consequently validate our technique for the memory determination in real data sequences modeled with Markov chains of arbitrary order.

#### 5. Conclusions

We have developed a novel method to determine the memory of a discrete sequence. Importantly, the method is valid for both Markovian



**Fig. 7.** Results of the method explained in Section 2 applied to sequences of data of daily precipitations for four locations: (a) Rome-Italy, (b) Dallas-USA, (c) Bangkok-Thailand and (d) Than Lwin-Myanmar.

and non-Markovian systems. Since the technique relies on the calculation of the Shannon entropy as a function of the block size, it is crucial to additionally propose an entropy estimator that gives good results for correlated systems. To this end, we have introduced a correction to the estimated probabilities to amend the source of error stemming from unseen elements in small samples. Our estimator is shown to significantly increase the accuracy of the entropy for systems with memory. Both numerically generated sequences and real data series have been used as benchmarks. These successful results will certainly encourage further applications of the proposals discussed in this work. It is left as a future project to present a more detailed comparison of our entropy estimator with different estimators that have shown promising results for independent sequences [36–39].

#### CRediT authorship contribution statement

**Juan De Gregorio:** Conceptualization, Methodology, Software, Writing – original draft, Writing – review & editing. **David Sánchez:** Conceptualization, Methodology, Writing – original draft, Writing – review & editing. **Raúl Toral:** Conceptualization, Methodology, Writing – original draft, Writing – review & editing.

#### Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

#### Data availability

No data was used for the research described in the article.

#### Acknowledgments

Partial financial support has been received from MCIN/AEI/10.13039/501100011033 and the Fondo Europeo de Desarrollo Regional (FEDER, UE) through projects PACSS (RTI2018-093732-B-C21), APASOS (PID2021-122256NB-C21) and the María de Maeztu Program for units of Excellence in R&D, grant MDM-2017-0711.

**Appendix A**

**A.1. Proof that if a sequence has memory  $m$ , then  $H_n$  is a linear function for  $n \geq m$**

Since we consider only homogeneous sequences we will omit the time variable in this appendix. We will use the notation  $p(x_1, \dots, x_s) \equiv P(X_1 = x_1, \dots, X_s = x_s)$ , with  $x_i \in \{z_j\}_{1 \leq j \leq L}$ .

Let us calculate the difference between the block entropies  $H_{n+1} - H_n$ . Using  $p(x_1, \dots, x_{n+1}) = p(x_{n+1}|x_1, \dots, x_n)p(x_1, \dots, x_n)$ , we obtain

$$\begin{aligned} H_{n+1} - H_n &= - \sum_{x_1, \dots, x_{n+1}} p(x_1, \dots, x_{n+1}) \log(p(x_1, \dots, x_{n+1})) + \sum_{x_1, \dots, x_n} p(x_1, \dots, x_n) \log(p(x_1, \dots, x_n)) \\ &= - \sum_{x_1, \dots, x_{n+1}} p(x_1, \dots, x_{n+1}) \log(p(x_{n+1}|x_1, \dots, x_n)) - \sum_{x_1, \dots, x_{n+1}} p(x_1, \dots, x_{n+1}) \log(p(x_1, \dots, x_n)) + \\ &+ \sum_{x_1, \dots, x_n} p(x_1, \dots, x_n) \log(p(x_1, \dots, x_n)). \end{aligned} \tag{A.1}$$

Because  $\sum_{x_{n+1}} p(x_1, \dots, x_{n+1}) = p(x_1, \dots, x_n)$ , the last two terms of the Eq. (A.1) cancel out. Thus,

$$H_{n+1} - H_n = - \sum_{x_1, \dots, x_{n+1}} p(x_1, \dots, x_{n+1}) \log(p(x_{n+1}|x_1, \dots, x_n)). \tag{A.2}$$

This is a general result, valid for any kind of sequence. Now, for a sequence of memory  $m$  and for  $n \geq m$ , one has  $p(x_{n+1}|x_1, \dots, x_n) = p(x_{n+1}|x_{n-m+1}, \dots, x_n)$ . Then, Eq. (A.2) becomes

$$\begin{aligned} H_{n+1} - H_n &= - \sum_{x_1, \dots, x_{n+1}} p(x_1, \dots, x_{n+1}) \log(p(x_{n+1}|x_{n-m+1}, \dots, x_n)) \\ &= - \sum_{x_{n-m+1}, \dots, x_{n+1}} p(x_{n-m+1}, \dots, x_{n+1}) \log(p(x_{n+1}|x_{n-m+1}, \dots, x_n)). \end{aligned} \tag{A.3}$$

For homogeneous sequences, we can shift all indices on the right-hand side of Eq. (A.3) by an amount  $n - m \geq 0$ :

$$H_{n+1} - H_n = - \sum_{x_1, \dots, x_{m+1}} p(x_1, \dots, x_{m+1}) \log(p(x_{m+1}|x_1, \dots, x_m)). \tag{A.4}$$

As far as  $n \geq m$ , the right-hand side of Eq. (A.4) is independent of  $n$ . Making the replacement  $n \rightarrow m$  we arrive at

$$H_{n+1} - H_n = H_{m+1} - H_m, \quad n \geq m, \tag{A.5}$$

which proves that the dependence of  $H_n$  on  $n$  is linear for  $n \geq m$ , i.e.,  $H_n = an + b$  with constant parameters  $a$  and  $b$ .

**A.2. Proof that if  $H_n$  is linear for  $n \geq m$ , then the sequence has memory  $m$**

Let  $H_n$  be linear for  $n \geq m$ . Then, we can write  $H_n = an + b$  or

$$H_{n+1} - H_n = H_{m+1} - H_m, \quad n \geq m. \tag{A.6}$$

Using the general result of Eq. (A.2) on the right-hand side of Eq. (A.6), we find

$$H_{n+1} - H_n = - \sum_{x_1, \dots, x_{m+1}} p(x_1, \dots, x_{m+1}) \log(p(x_{m+1}|x_1, \dots, x_m)). \tag{A.7}$$

For homogeneous sequences, we can shift all indices on the right-hand side of Eq. (A.7) by an amount  $n - m \geq 0$ :

$$\begin{aligned} H_{n+1} - H_n &= - \sum_{x_{n-m+1}, \dots, x_{n+1}} p(x_{n-m+1}, \dots, x_{n+1}) \log(p(x_{n+1}|x_{n-m+1}, \dots, x_n)) \\ &= - \sum_{x_1, \dots, x_{n+1}} p(x_1, \dots, x_{n+1}) \log(p(x_{n+1}|x_{n-m+1}, \dots, x_n)). \end{aligned} \tag{A.8}$$

We now apply Eq. (A.2) on the left-hand side of Eq. (A.8):

$$\sum_{x_1, \dots, x_{n+1}} p(x_1, \dots, x_{n+1}) \log(p(x_{n+1}|x_1, \dots, x_n)) = \sum_{x_1, \dots, x_{n+1}} p(x_1, \dots, x_{n+1}) \log(p(x_{n+1}|x_{n-m+1}, \dots, x_n)), \tag{A.9}$$

which implies that

$$\sum_{x_1, \dots, x_{n+1}} p(x_1, \dots, x_{n+1}) \log \left( \frac{p(x_{n+1}|x_1, \dots, x_n)}{p(x_{n+1}|x_{n-m+1}, \dots, x_n)} \right) = 0, \tag{A.10}$$

or

$$\sum_{x_1, \dots, x_n} p(x_1, \dots, x_n) \sum_{x_{n+1}} p(x_{n+1}|x_1, \dots, x_n) \log \left( \frac{p(x_{n+1}|x_1, \dots, x_n)}{p(x_{n+1}|x_{n-m+1}, \dots, x_n)} \right) = 0. \tag{A.11}$$

Because of the log sum inequality [26], we know that the second sum of Eq. (A.11) is  $\geq 0$  and generally  $p(x_1, \dots, x_n) > 0$ . Hence, Eq. (A.11) holds only if

$$\sum_{x_{n+1}} p(x_{n+1}|x_1, \dots, x_n) \log \left( \frac{p(x_{n+1}|x_1, \dots, x_n)}{p(x_{n+1}|x_{n-m+1}, \dots, x_n)} \right) = 0 \quad \forall x_1, \dots, x_n. \tag{A.12}$$

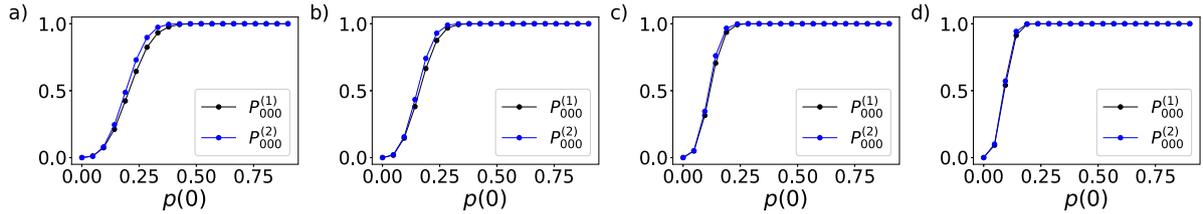
Further, due to the log sum inequality, Eq. (A.12) is valid provided that  $p(x_{n+1}|x_1, \dots, x_n) = p(x_{n+1}|x_{n-m+1}, \dots, x_n) \quad \forall x_{n+1}, x_1, \dots, x_n, \quad \forall n \geq m$ , which means that the sequence has memory  $m$ .

**Appendix B**

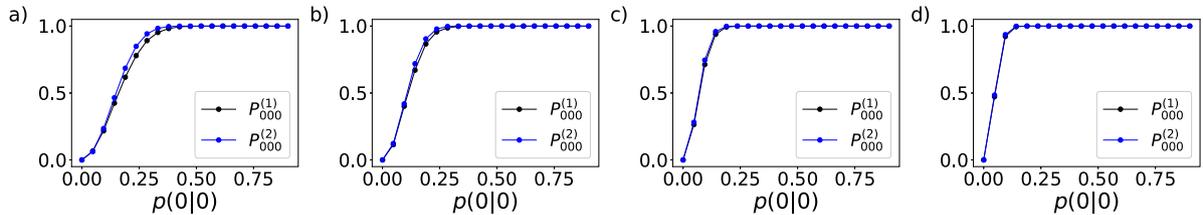
Given a sequence  $S$  of length  $N$ , we calculate the probability that we observe the block  $b_i^{(n)}$  of size  $n$  as follows,

$$P(b_i^{(n)} \in S) = 1 - (1 - p(b_i^{(n)}))^{Nn}. \tag{B.1}$$

Even though Eq. (B.1) is exact only when  $n = 1$  and the sequence memory is  $m = 0$ , we have checked in all our simulations that the corrections introduced by correlations when  $m > 0$  and  $n > 1$  can be neglected if  $N \gg n$ .



**Fig. B.1.**  $P_{000}^{(1)}$  (black lines) for binary sequences with memory  $m = 0$  as a function of the probability  $p(0)$  for occurrence of outcome 0. For each combination of  $N$  and  $p(0)$ , we generate  $K = 10^4$  numerical sequences and calculate  $P_{000}^{(1)}$  from  $K_{000}/K$ , where  $K_{000}$  is the number of sequences where the block  $(0, 0, 0)$  appears at least once. We also plot with blue lines  $P_{000}^{(2)} = 1 - (1 - (p(0))^3)^{N-2}$ . We show results for  $N = 100$  in (a), 200 in (b), 500 in (c) and 1000 in (d). We observe that the black and blue curves overlap as  $N$  grows.



**Fig. B.2.**  $P_{000}^{(1)}$  (black lines) for binary sequences with memory  $m = 1$  as a function of the conditional probability  $p(0|0)$ . For each combination of  $N$  and  $p(0|0)$ , we generate  $K = 10^4$  numerical sequences and calculate  $P_{000}^{(1)}$  from  $K_{000}/K$ , where  $K_{000}$  is the number of sequences where the block  $(0, 0, 0)$  appears at least once. We also plot with blue lines  $P_{000}^{(2)}$  calculated from the transition probabilities. For all the cases considered we take  $p(1|1) = 0.6$ . We show results for  $N = 100$  in (a), 200 in (b), 500 in (c) and 1000 in (d). We observe that the black and blue curves overlap as  $N$  grows.

As an example, we now assess the probability that the block  $(X_s = 0, X_{s+1} = 0, X_{s+2} = 0) \equiv (0, 0, 0)$  appears in  $S$  by generating  $K$  numerical sequences for fixed parameters  $L, m$  and  $N$ , and a particular set of transition probabilities. If the block  $(0, 0, 0)$  appears in  $K_{000}$  of those sequences, then  $P_{000}^{(1)} \equiv \hat{P}((0, 0, 0) \in S) = K_{000}/K$ . We note that for  $K \gg 1$   $P_{000}^{(1)} \simeq P((0, 0, 0) \in S)$ . We compare this result with the value  $P_{000}^{(2)} \equiv 1 - (1 - p(0, 0, 0))^{N-2}$ , where  $p(0, 0, 0)$  is the probability of occurrence of the block  $(0, 0, 0)$ , which is computed from the transition probabilities.

In Fig. B.1 we plot the results for both  $P_{000}^{(1)}$  and  $P_{000}^{(2)}$  obtained from for binary sequences with  $m = 0$ , as a function of  $p(0)$ . We consider the cases  $N = 100, 200, 500, 1000$  in Figs. B.1(a), (b), (c) and (d), respectively, for  $K = 10^4$  repetitions.

In Fig. B.2 we show similar curves but now considering sequences with memory  $m = 1$ . For each case, we fix  $p(1|1) = 0.6$  and vary  $p(0|0)$  between 0 and 0.9.

The results plotted in both Figs. B.1 and B.2 clearly show that as  $N$  increases the difference between  $P_{000}^{(1)}$  and  $P_{000}^{(2)}$  vanishes. We have verified with our simulations that this holds for every block sequence and for different values of  $m$ . Therefore, Eq. (B.1) is an excellent approximation when the size of the sequence is much larger than the size of the block, regardless of the memory value.

**Appendix C**

In Fig. C.1 we show the exact entropy  $H_n$  for binary systems with memory  $m = 2$  in (a) and  $m = 3$  in (b). In both cases the transition probabilities are chosen randomly from an uniform distribution  $(0, 1)$ . As a comparison, we also show the results obtained with the MLE estimator given by Eq. (8) (green line and dots), the Chao-Shen estimator given by Eq. (15) (blue line and dots) and our coverage-adjusted estimator proposed in Eq. (19) (red line and dots), all calculated from a sequence of  $N = 10^4$  data points generated numerically.

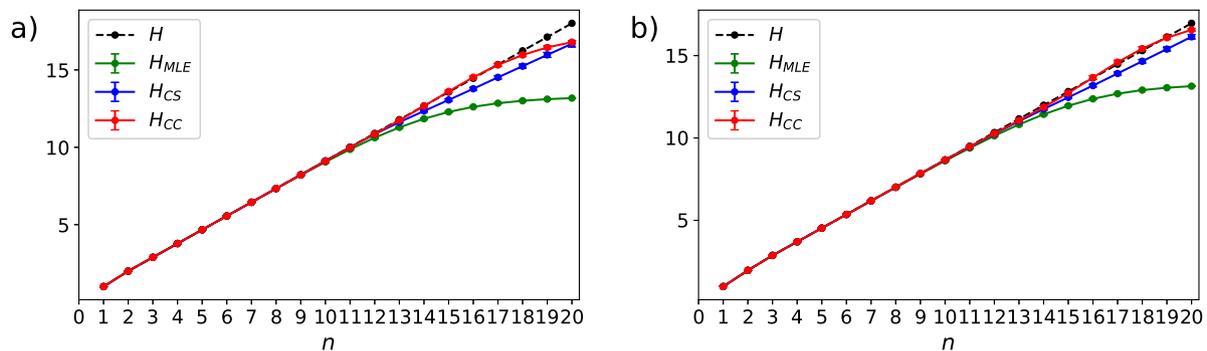


Fig. C.1. Exact Shannon entropy per block of size  $n$  (dotted line) for binary systems with  $m = 2$  in (a) and  $m = 3$  in (b) with fixed transition probabilities chosen randomly from a uniform distribution. A sequence of  $N = 10^4$  realizations is numerically generated from which we calculate three different entropy estimators discussed in the main text. The best performance is shown for the estimator that takes into account correlations (red dots).

## References

- [1] Cox DR, Miller HD. *The theory of stochastic processes*. Routledge; 2017.
- [2] Markov AA. An example of statistical investigation of the text eugene oegin concerning the connection of samples in chains. *Sci Context* 2006;19(4):591–600. <http://dx.doi.org/10.1017/S0269889706001074>.
- [3] Gardiner CW. *Handbook of stochastic methods for physics, chemistry and the natural sciences*. Berlin: Springer; 1965.
- [4] Churchill GA. Stochastic models for heterogeneous DNA sequences. *Bull Math Biol* 1989;51(1):79–94, URL <https://www.sciencedirect.com/science/article/pii/S0092824089800497>.
- [5] Wilks DS, Wilby RL. The weather generation game: a review of stochastic weather models. *Prog Phys Geogr: Earth Environ* 1999;23(3):329–57. <http://dx.doi.org/10.1177/030913339902300302>.
- [6] Page L, Brin S, Motwami R, Winograd T. *The PageRank citation ranking: Bringing order to the Web*. Technical report 1999–66, Stanford InfoLab; 1999.
- [7] Hänggi P, Thomas H. Stochastic processes: Time evolution, symmetries and linear response. *Phys Rep* 1982;88(4):207–319. [http://dx.doi.org/10.1016/0370-1573\(82\)90045-X](http://dx.doi.org/10.1016/0370-1573(82)90045-X).
- [8] Mayzelis Z, Apostolov S, Melnyk S, Usatenko O, Yampol'skii V. Additive N-step Markov chains as prototype model of symbolic stochastic dynamical systems with long-range correlations. *Chaos Solitons Fractals* 2007;34(1):112–28. <http://dx.doi.org/10.1016/j.chaos.2007.01.054>.
- [9] Raftery AE. A model for high-order Markov chains. *J R Stat Soc Ser B Stat Methodol* 1985;47(3):528–39. <http://dx.doi.org/10.1111/j.2517-6161.1985.tb01383.x>.
- [10] Yulmetyev RM, Demin SA, Panishev OY, Hänggi P, Timashev SF, Vstovsky GV. Regular and stochastic behavior of parkinsonian pathological tremor signals. *Physica A* 2006;369(2):655–78. <http://dx.doi.org/10.1016/j.physa.2006.01.077>.
- [11] Ho DT, Cao TH. A high-order hidden Markov model for emotion detection from textual data. In: *Pacific Rim knowledge acquisition workshop*. Springer; 2012, p. 94–105.
- [12] Seifert M, Gohr A, Strickert M, Grosse I. Parsimonious higher-order hidden Markov models for improved array-CGH analysis with applications to Arabidopsis Thaliana. *PLoS Comput Biol* 2012;8(1):e1002286. <http://dx.doi.org/10.1371/journal.pcbi.1002286>.
- [13] Singer P, Helic D, Taraghi B, Strohmaier M. Detecting memory and structure in human navigation patterns using Markov chain models of varying order. *PLoS One* 2014;9(7):e102070. <http://dx.doi.org/10.1371/journal.pone.0114952>.
- [14] Meyer H, Rieger H. Optimal non-Markovian search strategies with  $n$ -step memory. *Phys Rev Lett* 2021;127:070601. <http://dx.doi.org/10.1103/PhysRevLett.127.070601>.
- [15] Streliaoff CC, Crutchfield JP, Hübler AW. Inferring Markov chains: Bayesian estimation, model comparison, entropy rate, and out-of-class modeling. *Phys Rev E* 2007;76:011106. <http://dx.doi.org/10.1103/PhysRevE.76.011106>.
- [16] Akaike H. Information theory and an extension of the maximum likelihood principle. In: *Selected papers of Hirotugu Akaike*. Springer; 1998, p. 199–213.
- [17] Tong H. Determination of the order of a Markov chain by Akaike's information criterion. *J Appl Probab* 1975;12(3):488–97. <http://dx.doi.org/10.2307/3212863>.
- [18] Katz RW. On some criteria for estimating the order of a Markov chain. *Technometrics* 1981;23(3):243–9. <http://dx.doi.org/10.2307/1267787>.
- [19] Weakliem DL. A critique of the Bayesian information criterion for model selection. *Sociol Methods Res* 1999;27(3):359–97. <http://dx.doi.org/10.1177/0049124199027003002>.
- [20] Schmitt AO, Herzel H. Estimating the entropy of DNA sequences. *J Theoret Biol* 1997;188(3):369–77. <http://dx.doi.org/10.1006/jtbi.1997.0493>.
- [21] Chao A, Shen T-J. Nonparametric estimation of Shannon's diversity index when there are unseen species in sample. *Environ Ecol Stat* 2003;10:429–43. <http://dx.doi.org/10.1023/A:1026096204727>.
- [22] Häggström O, et al. *Finite Markov chains and algorithmic applications*, Vol. 52. Cambridge University Press; 2002.
- [23] Pompe B. On some entropy methods in data analysis. *Chaos Solitons Fractals* 1994;4(1):83–96. [http://dx.doi.org/10.1016/0960-0779\(94\)90019-1](http://dx.doi.org/10.1016/0960-0779(94)90019-1).
- [24] Shannon CE. A mathematical theory of communication. *Bell Syst Tech J* 1948;27(3):379–423. <http://dx.doi.org/10.1002/j.1538-7305.1948.tb01338.x>.
- [25] Park S, Pande VS. Validation of Markov state models using Shannon's entropy. *J Chem Phys* 2006;124(5):054118. <http://dx.doi.org/10.1063/1.2166393>.
- [26] Cover T, Thomas J. *Elements of information theory*. John Wiley and Sons; 2006.
- [27] Schürmann T. Bias analysis in entropy estimation. *J Phys A: Math Gen* 2004;37(27):L295–301. <http://dx.doi.org/10.1088/0305-4470/37/27/102>.
- [28] Paninski L. Estimation of entropy and mutual information. *Neural Comput* 2003;15(6):1191–253. <http://dx.doi.org/10.1162/08997660321780272>.
- [29] Contreras Rodríguez L, Madarro-Capó EJ, Legón-Pérez CM, Rojas O, Sosa-Gómez G. Selecting an effective entropy estimator for short sequences of bits and bytes with maximum entropy. *Entropy* 2021;23(5). <http://dx.doi.org/10.3390/e23050561>.
- [30] Horvitz DG, Thompson DJ. A generalization of sampling without replacement from a finite universe. *J Amer Statist Assoc* 1952;47(260):663–85. <http://dx.doi.org/10.1080/01621459.1952.10483446>.
- [31] Good I. The population frequencies of species and the estimation of population parameters. *Biometrika* 1953;40:237–64. <http://dx.doi.org/10.2307/2333344>.
- [32] Wan H, Zhang X, Barrow EM. Stochastic modelling of daily precipitation for Canada. *Atmos-Ocean* 2005;43(1):23–32. <http://dx.doi.org/10.3137/ao.430102>.
- [33] Wilson Kemsley S, Osborn TJ, Dorling SR, Wallace C, Parker J. Selecting Markov chain orders for generating daily precipitation series across different Köppen climate regimes. *Int J Climatol* 2021;41:6223–37. <http://dx.doi.org/10.1002/joc.7175>.
- [34] Schwarz G. Estimating the dimension of a model. *Ann Statist* 1978;6(2):461–4. <http://dx.doi.org/10.1214/aos/1176344136>.
- [35] Menne MJ, Durre I, Vose RS, Gleason BE, Houston TG. An overview of the global historical climatology network-daily database. *J Atmos Ocean Technol* 2012;29(7):897–910. <http://dx.doi.org/10.1175/JTECH-D-11-00103.1>.
- [36] Nemenman I, Shafee F, Bialek W. Entropy and inference, revisited. In: Di-etterich T, Becker S, Ghahramani Z, editors. *Advances in neural information processing systems*, Vol. 14. MIT Press; 2001, URL <https://proceedings.neurips.cc/paper/2001/file/d46e1fc4c07ce4a69ee07e4134bcecf-Paper.pdf>.
- [37] Archer EW, Park IM, Pillow JW. Bayesian entropy estimation for binary spike train data using parametric prior knowledge. In: Bourges C, Bottou L, Welling M, Ghahramani Z, Weinberger K, editors. *Advances in neural information processing systems*, Vol. 26. Curran Associates, Inc.; 2013, URL <https://proceedings.neurips.cc/paper/2013/file/f9a40a4780f5e1306c46f1c8daeece3b-Paper.pdf>.
- [38] Hausser J, Strimmer K. Entropy inference and the James-Stein Estimator, with application to nonlinear gene association networks. *J Mach Learn Res* 2009;10(50):1469–84, URL <http://jmlr.org/papers/v10/hausser09a.html>.
- [39] Grassberger P. On generalized Schürmann entropy estimators. *Entropy* 2022;24(5):680. <http://dx.doi.org/10.3390/e24050680>.