AMB ALGORITHMS FOR MOLECULAR BIOLOGY

RESEARCH Open Access

Exact distribution of a pattern in a set of random sequences generated by a Markov source: applications to biological data

Gregory Nuel[1,2,3*], Leslie Regad[4,5†], Juliette Martin[4,6,7†], Anne-Claude Camproux[4,5]

Abstract

Background: In bioinformatics it is common to search for a pattern of interest in a potentially large set of rather short sequences (upstream gene regions, proteins, exons, etc.). Although many methodological approaches allow practitioners to compute the distribution of a pattern count in a random sequence generated by a Markov source, no specific developments have taken into account the counting of occurrences in a set of independent sequences. We aim to address this problem by deriving efficient approaches and algorithms to perform these computations both for low and high complexity patterns in the framework of homogeneous or heterogeneous Markov models.

Results: The latest advances in the field allowed us to use a technique of optimal Markov chain embedding based on deterministic finite automata to introduce three innovative algorithms. Algorithm 1 is the only one able to deal with heterogeneous models. It also permits to avoid any product of convolution of the pattern distribution in individual sequences. When working with homogeneous models, Algorithm 2 yields a dramatic reduction in the complexity by taking advantage of previous computations to obtain moment generating functions efficiently. In the particular case of low or moderate complexity patterns, Algorithm 3 exploits power computation and binary decomposition to further reduce the time complexity to a logarithmic scale. All these algorithms and their relative interest in comparison with existing ones were then tested and discussed on a toy-example and three biological data sets: structural patterns in protein loop structures, PROSITE signatures in a bacterial proteome, and transcription factors in upstream gene regions. On these data sets, we also compared our exact approaches to the tempting approximation that consists in concatenating the sequences in the data set into a single sequence.

Conclusions: Our algorithms prove to be effective and able to handle real data sets with multiple sequences, as well as biological patterns of interest, even when the latter display a high complexity (PROSITE signatures for example). In addition, these exact algorithms allow us to avoid the edge effect observed under the single sequence approximation, which leads to erroneous results, especially when the marginal distribution of the model displays a slow convergence toward the stationary distribution. We end up with a discussion on our method and on its potential improvements.

Introduction

The availability of biological sequence data prior to any kinds of data is one of the major consequences of the revolution brought by high throughput biology. Large-scale DNA sequencing projects now routinely produce huge amounts of DNA sequences, and the protein sequences deduced from them. The number of completely sequenced genomes stored in the Genome Online Database [1] has already reached the impressive number of 2, 968. Currently, there are about 99 million DNA sequences in Genbank [2] and 8.6 million proteins in the UniProtKB/TrEMBL database [3]. Sequence analysis has become a major field of bioinformatics, and it is now natural to search for patterns (also called motifs) in biological sequences. Sequence patterns in biological sequences can have functional or structural implications such as promoter regions or transcription factor binding sites in DNA, or functional family signature in proteins.

* Correspondence: gregory.nuel@parisdescartes.fr
† Contributed equally
[1]LSG, Laboratoire Statistique et Génome, CNRS UMR-8071, INRA UMR-1152, University of Evry, Evry, France

Because they are important for function or structure, such patterns are expected to be subject to positive or negative selection pressures during evolution, and consequently they appear more or less frequently than expected. This assumption has been used to search for exceptional words in a particular genome [4,5]. Another successful application of this approach is the identification of specific functional patterns: restriction sites [6], cross-over hotspot instigator sites [7], polyadenylation signals [8], etc. Obviously the results of such an approach strongly depend on the biological relevance of the data set used. A convenient way to discover these patterns is to build multiple sequence alignments, and look for conserved regions. This is done, for example, in the PROSITE database, a dictionary of functional signatures in protein sequences [9]. However, it is not always possible to produce a multiple sequence alignment.

In this paper, patterns refer to a finite family of words (or a regular expression), which is a slightly different notion from that of Position Specific Scoring Matrices (PSSM) [10] or in a similar way, from Position Weighted Matrices (PWM) or HMM profiles. Indeed, PSSM provide a scoring scheme to scan any sequence for possible occurrence of a given signal. When one defines a pattern ocurrence as a position where the PSSM score is above a given threshold, it is possible to associate a regular expression to this particular pattern. In that sense, PSSM may be seen as a particular case of the class of patterns we considered in this paper. However, this approach usually leads to huge regular expressions whose complexity grows geometrically with the PSSM length. For that reason, it seems far more efficient to deal with PSSM problems with methods and techniques that have been specifically developed for them [11,12].

Pattern statistics offer a convenient framework to treat non-aligned sequences, as well as assessing the statistical significance of patterns. It is also a way to discover putative functional patterns from whole genomes using statistical exceptionality. In their pioneer study, Karlin et al. investigated 4- and 6-palindromes in DNA sequences from a broad range of organisms, and found that these patterns had significantly low counts in bacteriophages, probably as a means of avoiding restriction enzyme cleavage by the host bacteria [6]. Then they analyzed the statistical over- or under-representation of short DNA patterns in herpes viruses using z-scores and Markov models, and used them to construct an evolutionary tree [4]. In another study, the authors analyzed the genome of *Bacillus subtilis* and found a large number of words of length up to 8 nucleotides with biased representation [5]. Another striking example of functional patterns with unusual frequency is the Chi motif (cross-over hot-spot instigator site) in *Escherichia coli* [7]. Pattern statistics have also been used to detect putative polyadenylation signals in yeast [8].

In general, patterns with unusual frequency are detected by comparing their observed frequency in the biological sequence data under study to their distribution in a background model whose parameters are derived from the data. Among a wide range of possible models, a popular choice consists in considering only homogeneous Markov models of fixed order. This choice is motivated both by the fact that the statistical properties of such models are well known, and that it is a very natural way to take into account the sequence bias in letters (order 0 Markov model), or words of size $h \geq 2$ (order h - 1 Markov model). However, it is well-known that biological sequences usually display high heterogeneity. Genome sequences, for example, are intrinsically heterogeneous, across genomes as well as between regions in the same genome [13]. In their study of the *Bacillus subtilis* chromosome, Nicolas et al. identified different compositional classes using a hidden Markov model [14]. These different compositional classes showed a good correspondence with coding and non-coding regions, horizontal gene transfer, hydrophobic protein coding regions and highly expressed genes. DNA heterogeneity is indeed used for gene prediction [15] and horizontal transfer detection [16]. Protein sequences also display sequence heterogeneity. For example, the amino-acid composition differs according to the secondary structure (alpha-helix, beta-strand and loop), and this property has also been used to predict the secondary structure from the amino-acid sequence using hidden Markov models [17]. In order to take into account this natural heterogeneity of biological data, it is common to assume either that the data are piecewise homogeneous (that is typically what is done with hidden Markov models [18]), or simply that the model changes continuously from one position to another (e. g., walking Markov models [19]). One should note that such fully heterogeneous models may also appear naturally as the consequences of a previous modeling attempt [20,21].

A biological pattern study usually first consists in gathering a data set of sequences sharing similar features (ribosome binding sites, related protein domains, donor or acceptor sites in eucaryotic DNA, secondary or tertiary structures of proteins, etc.). The resulting data set typically contains a large number of rather short sequences (ex: 5,000 sequences of lengths ranging between 20 and 300). Then one searches this data set for patterns that occur much more (or less) than expected under the null model. The goal of this paper is to provide efficient algorithms to assess the statistical significance of patterns both for low and high

complexity patterns in sets of multiple sequences generated by homogeneous or heterogeneous Markov sources.

From the statistical point of view, studying the distribution of the random count of a simple or complex pattern in a multi-state homogeneous or heterogenous Markov chain is a difficult task. A lot of effort has gone into tackling this problem in the last fifty years with many concurrent approaches and here we give only a few references; see [22-25] for a more comprehensive review. Exact methods are based on a wide range of techniques like Markov chain embedding, moment generating functions, combinatorial methods, or exponential families [26-33]. There is also a wide range of asymptotic approximations, the most popular of which are Gaussian approximations [34-37], Poisson approximations [38-42] and Large Deviation approximations [43-45].

Recently several authors [46-49] have pointed out the connexion between the distribution of random pattern counts in Markov chains and the pattern matching theory. Thanks to these approaches, it is now possible to obtain an optimal Markov chain embedding of any pattern problem through minimal Deterministic Finite Automata (DFA).

In this paper, we first recall the technique of optimal Markov chain embedding for pattern problems and how it allows obtaining the distribution of a pattern count in the particular case when a single sequence is considered. We then extend this result to a set of several sequences and provide three efficient algorithms to cover the practical computation of the corresponding distribution, either for heterogeneous or homogeneous models, and patterns of various complexity. In the second part of the paper, we apply our methods to a simple but illustrative toy-example, and then consider three real-life biological applications: structural patterns in protein loop structures, PROSITE signatures in a bacteria proteome, and transcription factors in upstream gene regions. Finally, the results, methods and possible improvements are discussed.

Methods

Model and notations

Let $(X_i)_{1\leq i\leq \ell}$ be an order $d \geq 0$ Markov chain over the finite alphabet $\mathcal{A}$ (with cardinal $|\mathcal{A}| \geq 2$). For all $1 \leq i \leq j \leq \ell$, we denote by $X_j^i \overset{\text{def}}{=} X_i \dots X_j$ the subsequence between positions i and j. For all $a_1^d \overset{\text{def}}{=} a_1 \dots a_d \in \mathcal{A}^d$, $b \in \mathcal{A}$, and $1 \leq i \leq \ell - d$, let us denote by $\mu(a_1^d) \overset{\text{def}}{=} \mathbb{P}(X_1^d = a_1^d)$ the starting distribution and by $\pi_{i+d}(a_1^d, b) \overset{\text{def}}{=} \mathbb{P}(X_{i+d} = b \mid X_i^{i+d-1} = a_1^d)$ the transition probability towards X_{i+d}.

Let $\mathcal{W}$ be a finite set of words (for simplification purpose, we assume that $\mathcal{W}$ contains no word of length less than d - in the general case, one may have to count the pattern occurrences already seen in X_1^d, which results in a more complex starting distribution for our embedding Markov chain) over $\mathcal{A}$. We consider the random number N_ℓ of matching positions of $\mathcal{W}$ in X_1^ℓ defined by:

$$N_\ell \overset{\text{def}}{=} \sum_{i=1}^{\ell} \mathbb{I}_{\{\mathcal{W} \cap \mathcal{S}(X_1^i) \neq \varnothing\}} \tag{1}$$

where $\mathcal{S}(X_1^i)$ is the set of all the suffixes of X_1^i and where $\mathbb{I}_A$ is the indicator function of event A.

Overview of the Markov chain embedding

As suggested in [46-49], we perform an optimal Markov chain embedding of our pattern problem through a DFA. We use here the notations of [49]. Let $(\mathcal{A}, \mathcal{Q}, \sigma, \mathcal{F}, \delta)$ be a *minimal* DFA that recognizes the language $\mathcal{W}^*\mathcal{W}$ of all texts over $\mathcal{A}$ ending with an occurrence of $\mathcal{W}$ where $\mathcal{A}^*$ denotes the set of all - possibly empty - texts over $\mathcal{A}$. $\mathcal{Q}$ is a finite state space, $\sigma \in \mathcal{Q}$ is the starting state, $\mathcal{F} \subset \mathcal{Q}$ is the subset of final states and $\delta : \mathcal{Q} \times \mathcal{A} \to \mathcal{Q}$ is the transition function. We recursively extend the definition of δ over $\mathcal{Q} \times \mathcal{A}^*$ thanks to the relation $\delta(p, aw) \overset{\text{def}}{=} \delta(\delta(p, a), w)$ for all $p \in \mathcal{Q}$, $a \in \mathcal{A}$, $w \in \mathcal{A}^*$. We additionally suppose that this automaton is non d-ambiguous (a DFA having this property is also called a d-th order DFA in [48]), which means that for all $q \in \mathcal{Q}$, the set $\delta^{-d}(q) \overset{\text{def}}{=} \{a_1^d \in \mathcal{A}^d, \exists p \in \mathcal{Q}, \delta(p, a_1^d) = q\}$ of sequences of length d that can lead to q is either a singleton or the empty set. A DFA is hence said to be non d-ambiguous if the past of order d is uniquely defined for all states. When the notation is not ambiguous, the set $\delta^{-d}(q)$ may also denote its unique element (singleton case).

Theorem 1. We consider the random sequence over $\mathcal{Q}$ defined by $\tilde{X}_0 \overset{\text{def}}{=} \sigma$ and $\tilde{X}_i \overset{\text{def}}{=} \delta(\tilde{X}_{i-1}, X_i) \forall i, 1 \leqslant i \leqslant \ell$. Then $(\tilde{X}_i)_{i\geq d}$ is a heterogeneous order 1 Markov chain over $\mathcal{Q}' \overset{\text{def}}{=} \delta(\sigma, \mathcal{A}^d\mathcal{A}^*)$ such that, for all $p, q \in \mathcal{Q}'$ and $1 \leq i \leq \ell - d$ the starting distribution $\mathbf{m}_d(p) \overset{\text{def}}{=} \mathbb{P}(\tilde{X}_d = p)$ and the transition matrix $\mathbf{T}_{i+d}(p, q) \overset{\text{def}}{=} \mathbb{P}(\tilde{X}_{i+d} = q \mid \tilde{X}_{i+d-1} = p)$ are given by:

$$\mathbf{m}_d(p) = \begin{cases} \mu(\delta^{-d}(p)) & \text{if } \delta^{-d}(p) \neq \varnothing \\ 0 & \text{otherwise} \end{cases}; \tag{2}$$

$$\mathbf{T}_{i+d}(p,q) = \begin{cases} \pi_{i+d}(\delta^{-d}(p), b) & \text{if } \exists b \in \mathcal{A}, \delta(p,b) = q \\ 0 & \text{otherwise} \end{cases} \quad (3)$$

And for all $i \geq d$ we have:

$$\mathcal{W} \cap \mathcal{S}(X_1^i) \neq \varnothing \Leftrightarrow \tilde{X}_i \in \mathcal{F}. \quad (4)$$

Proof. The result is immediate considering the properties of the DFA. See [48] or [49] for more details. □

From now on, we will denote the cardinality of the set $\mathcal{Q}'$ by L and call this the pattern complexity (even if technically, L depends both on the considered pattern and the Markov model order). A typical low complexity pattern corresponds to $L \leq 50$, moderate complexity to $50 < L < 100$, and high complexity to $L \geq 100$.

Proposition 2. The moment generating function $G_{N_\ell}(y)$ of N_ℓ is given by:

$$G_{N_\ell}(y) \overset{\text{def}}{=} \sum_{n=0}^{+\infty} \mathbb{P}(N_\ell = n) y^n = \mathbf{m}_d \left(\prod_{i=1}^{\ell-d} (\mathbf{P}_{i+d} + y\mathbf{Q}_{i+d}) \right) \mathbf{1}^{\mathrm{T}} \quad (5)$$

where $\mathbf{1}$ is a row vector of ones, and $\mathbf{1}^{\mathrm{T}}$ denotes the transpose vector, and, for all $1 \leq i \leq \ell - d$, $\mathbf{T}_{i+d} = \mathbf{P}_{i+d} + \mathbf{Q}_{i+d}$ with $\mathbf{P}_{i+d}(p,q) \overset{\text{def}}{=} \mathbb{I}_{q \notin \mathcal{F}} \mathbf{T}_{i+d}(p,q)$ and $\mathbf{Q}_{i+d}(p,q) \overset{\text{def}}{=} \mathbb{I}_{q \in \mathcal{F}} \mathbf{T}_{i+d}(p,q)$ for all $p, q \in \mathcal{Q}'$.

Proof. Since $\mathbf{Q}_{i+d}$ contains all the counting transitions, we keep track of the number of occurrences by associating a dummy variable y to these transitions. Therefore, we just have to compute the marginal distribution at the end of the sequence and sum up the contribution of each state. See [46-49] for more details. □

Corollary 3. In the particular case where $(X_i)_{1 \leq i \leq \ell}$ is a homogeneous Markov chain, we can drop the indices in $\mathbf{P}_{i+d}$ and $\mathbf{Q}_{i+d}$ and Equation (5) is simplified into

$$G_{N_\ell}(y) = \mathbf{m}_d (\mathbf{P} + y\mathbf{Q})^{\ell-d} \mathbf{1}^{\mathrm{T}}. \quad (6)$$

Corollary 3 can be found explicitly in [48] or [50] and its generalisation to a heterogeneous model (Proposition 2) is given in [51].

Extension to a set of sequences

Let us now assume that we consider a set of r sequences. For any particular sequence j (with $1 \leq j \leq r$) we denote by ℓ_j its length, by N_{ℓ_j} its number of pattern occurrences, and by $\mathbf{m}_d^j$, $\mathbf{P}_{i+d}^j$, and $\mathbf{Q}_{i+d}^j$ its corresponding Markov chain embedding parameters.

Proposition 4. If we denote by

$$G_N(y) \overset{\text{def}}{=} \sum_{n=0}^{+\infty} \mathbb{P}(N_{\ell_1} + \ldots + N_{\ell_r} = n) y^n \quad (7)$$

the moment generating function of $N \overset{\text{def}}{=} N_{\ell_1} + \ldots + N_{\ell_r}$, we have:

$$G_N(y) = \underbrace{\mathbf{m}_d^1 \left(\prod_{i=1}^{\ell_1-d} (\mathbf{P}_{i+d}^1 + y\mathbf{Q}_{i+d}^1) \right) \mathbf{1}^{\mathrm{T}}}_{G_{N_{\ell_1}}(y)} \times \ldots \times \underbrace{\mathbf{m}_d^r \left(\prod_{i=1}^{\ell_r-d} (\mathbf{P}_{i+d}^r + y\mathbf{Q}_{i+d}^r) \right) \mathbf{1}^{\mathrm{T}}}_{G_{N_{\ell_r}}(y)}. \quad (8)$$

Corollary 5. In the homogeneous case we get:

$$G_N(y) = \underbrace{\mathbf{m}_d^1 (\mathbf{P} + y\mathbf{Q})^{\ell_1-d} \mathbf{1}^{\mathrm{T}}}_{G_{N_{\ell_1}}(y)} \times \ldots \times \underbrace{\mathbf{m}_d^r (\mathbf{P} + y\mathbf{Q})^{\ell_r-d} \mathbf{1}^{\mathrm{T}}}_{G_{N_{\ell_r}}(y)}. \quad (9)$$

Single sequence approximation

Instead of computing the exact distribution of $N = N_1 + \ldots + N_r$, which requires specific developments, one may study the number N' of pattern occurrences in a single sequence of length $\ell = \ell_1 + \ldots + \ell_r$ resulting from the concatenation of our r sequences. The main advantage of this method is that we can rely on a wide range of classical techniques to compute the exact or approximated distribution of N' (Poisson approximation or large deviations for example).

The drawback of this approach is that N and N' are clearly two different random variables and that deriving the P-value of an observed event for N using the distribution of N' may produce erroneous results due to edge effects.

These effects may be caused by two distinct phenomena: forbidden positions and stationary assumption. Forbidden positions simply come from the fact that the artificial concatenated sequence may have pattern occurrences at positions that overlap two individual sequences. If we consider a pattern of length h, it is clear that there are $h - 1$ positions that overlap two sequences. It is hence natural to correct this effect by introducing an offset for each sequence, typically set to $h - 1$ for a pattern of length h. The length of our concatenated sequence has then to be adjusted to $\ell' = (\ell_1 - \text{offset}) + \ldots + (\ell_{r-1} - \text{offset}) + \ell_r = \ell - (r - 1) \times \text{offset}$. One should note that there is no canonical choice of offset for patterns of variable lengths.

Even if we take into account the forbidden overlapping positions with a proper choice of offset, there is a second phenomenon that may affect the quality of the single sequence approximation, and it is connected to the model itself. When one works with a single sequence, it is common to assume that the underlying model is stationary. This assumption is usually considered to be harmless since the marginal distribution of any non-stationary model converges very quickly towards its stationary distribution. As long as the time to convergence is negligible in comparison with the total length of the sequence, this approximation has a very small impact on the distribution. In the case where

we consider a data set composed of a large number of relatively short sequences, this edge effect might however have huge consequences. This obviously depends both on the difference between the starting distribution of the sequences, and on the convergence rate toward the stationary distribution. This phenomenon is studied in detail in our applications.

Algorithms

Let n be the observed number of occurrences of our pattern of interest. Our main objective is to compute both $\mathbb{P}(N \leq n)$ and $\mathbb{P}(N \geq n)$. We provide here various algorithms to perform these computations both for low or high complexity patterns, and for homogeneous or heterogenous models.

Heterogeneous case

Algorithm 1: Compute $\mathcal{T}_{n+1}(G_N(\gamma))$ (see Equation (10) for a proper definition of $\mathcal{T}_{n+1}$) in the case of a heterogeneous model. The workspace complexity is $O(n \times L)$ and since all matrix vector products exploit the sparse structure of the matrices, the time complexity is $O(\ell \times n \times |\mathcal{A}| \times L)$ where $|\mathcal{A}| \times L$ corresponds to the maximum number of non-zero terms in $\mathbf{T}_{i+d} = \mathbf{P}_{i+d} + \mathbf{Q}_{i+d}$.

Require: The starting distributions $\mathbf{m}_d^j$ the matrices $\mathbf{Q}_{i+d}^j$, $\mathbf{Q}_{i+d}^j$, for all $1 \leq j \leq r$, $1 \leq i \leq \ell_j - d$, a $O(n \times L)$ workspace to keep the current values of $\mathbf{E}(\gamma)$, and a dimension L polynomial row-vector of degree $n + 1$.

// *Initialization*
$\mathbf{E}(\gamma) \leftarrow \mathbf{1}$
// *Loop on sequences*
for $j = 1, ..., r$ **do**
  $\mathbf{E}(\gamma) \leftarrow (\mathbf{E}(\gamma)\mathbf{1}^{\mathrm{T}}) \times \mathbf{m}_d^j$
  // *Loop on positions within the sequence*
  for $i = 1, ... \ell_j - d$ **do**
    $\mathbf{E}(\gamma) \leftarrow \mathcal{T}_{n+1}\left(\mathbf{E}(\gamma) \times (\mathbf{P}_{i+d}^j + \gamma \mathbf{Q}_{i+d}^j)\right)$
Output: return $\mathcal{T}_{n+1}(G_N(\gamma)) = \mathbf{E}(\gamma)\mathbf{1}^{\mathrm{T}}$

When working with heterogeneous models, there is very little room for optimization in the computation of Equation (8). Indeed, since all terms $\mathbf{P}_{i+d}^j$ and $\mathbf{Q}_{i+d}^j$ may differ for each combination of position i and sequence j, there is no choice but to compute the individual contribution of each of these combinations. This may be done recursively by taking advantage of the sparsity of matrices $\mathbf{P}_{i+d}^j$ and $\mathbf{Q}_{i+d}^j$. Note that, so as to speed up the computation, it is not necessary to keep track of the polynomial terms of degrees greater than $n + 1$. This may be done by using the polynomial truncation function $\mathcal{T}_{n+1}$ defined by

$$\mathcal{T}_{n+1}\left(\sum_{k\geqslant 0} p_k \gamma^k\right) \stackrel{\text{def}}{=} \sum_{k=0}^{n} p_k \gamma^k + \left(\sum_{k>n} p_k\right)\gamma^{n+1}. \qquad (10)$$

This function also applies to vector or matrix polynomials. This approach results in Algorithm 1 whose time complexity is $O(\ell \times n \times |\mathcal{A}| \times L)$. In particular, one observes that the time complexity remains linear with n, which is a unique feature of this algorithm, while an individual computation of each $G_{N_{\ell_j}}(\gamma)$ would obviously result in a final $O(r \times n^2)$ complexity to perform the polynomial product $G_N(\gamma) = G_{N_{\ell_1}}(\gamma) \times \ldots \times G_{N_{\ell_r}}(\gamma)$. It is also interesting to point out that the number r of considered sequences does not appear explicitly in the complexity of Algorithm 1 but only through the total length $\ell \stackrel{\text{def}}{=} \ell_1 + \ldots + \ell_r$.

Homogeneous case

Algorithm 2: Compute the $\mathcal{T}_{n+1}(G_N(\gamma))$ in the case of a homogeneous model. The workspace complexity is $O(n \times L)$ and since all matrix vector products exploit the sparse structure of the matrices, the time complexity to compute all $\mathcal{T}_{n+1}(G_{N_{\ell_j}}(\gamma))$ is $O(\ell_r \times n \times |\mathcal{A}| \times L)$ where $|\mathcal{A}| \times L$ corresponds to the maximum number of non-zero terms in $\mathbf{T} = \mathbf{P} + \mathbf{Q}$. The product updates of $U(\gamma)$ result in a additional time complexity of $O(r \times n^2)$.

Require: The matrices $\mathbf{P}$ and $\mathbf{Q}$, for all $1 \leq j \leq r$, the starting distributions $\mathbf{m}_d^j$, the length ℓ_j (assuming $\ell_0 \stackrel{\text{def}}{=} d \leqslant \ell_1 \leqslant \ldots \leqslant \ell_r$), a $O(n \times L)$ workspace to keep the current values of $\mathbf{E}(\gamma)$ (a dimension L polynomial row-vector of degree $n + 1$) and $U(\gamma)$ (a polynomial of degree $n + 1$).

// *Initialization*
$U(\gamma) \leftarrow 1$ and $\mathbf{E}(\gamma) \leftarrow \mathbf{1}$
// *Loop on sequences*
for $j = 1, ..., r$ **do**
  for $i = 1, ..., \ell_j - \ell_{j-1}$ **do**
    $\mathbf{E}(\gamma)^{\mathrm{T}} \leftarrow \mathcal{T}_{n+1}((\mathbf{P} + \gamma\mathbf{Q})\mathbf{E}(\gamma)^{\mathrm{T}})$
  optionally return $\mathcal{T}_{n+1}(G_{N_{\ell_j}}(\gamma)) = \mathbf{m}_d^j \mathbf{E}(\gamma)^{\mathrm{T}}$
  $U(\gamma) \leftarrow \mathcal{T}_{n+1}\left(U(\gamma) \times \mathbf{m}_d^j \mathbf{E}(\gamma)^{\mathrm{T}}\right)$
Output: return $\mathcal{T}_{n+1}(G_N(\gamma)) = U(\gamma)$

If we now consider a homogeneous model, we can dramatically speed up the computation of Equation (9) by recycling intermediate results in order to compute efficiently all $G_{N_{\ell_j}}(\gamma)$. Without loss of generality, we assume that the sequences are ordered by increasing lengths: $\ell_1 \leq \ldots \leq \ell_r$. If one stores the value of $(\mathbf{P} + \gamma\mathbf{Q})^{\ell_1 - d}\mathbf{1}^{\mathrm{T}}$ in some polynomial vector $\mathbf{E}(\gamma)^{\mathrm{T}}$, it is clear that $(\mathbf{P} + \gamma\mathbf{Q})^{\ell_2 - d}\mathbf{1}^{\mathrm{T}} = (\mathbf{P} + \gamma\mathbf{Q})^{\ell_2 - \ell_1} \times \mathbf{E}(\gamma)^{\mathrm{T}}$. By repeating this trick for all ℓ_j, it is then possible to adapt Algorithm 1 to compute all $G_{N_{\ell_j}}$ with a complexity $O(\ell_r \times n \times |\mathcal{A}| \times L)$ (ℓ_r being the length of the longest sequence), which is a dramatic improvement. Unfortunately, it is then necessary to compute the product $G_N(\gamma) = G_{N_{\ell_1}}(\gamma) \times \ldots \times G_{N_{\ell_r}}(\gamma)$, which results in a complexity $O(r \times n^2)$ to get all polynomial terms of degree smaller that $n + 1$ in $G_N(\gamma)$. This additional complexity therefore limits the interest of this algorithm in comparison to Algorithm 1, especially when one observes a large number n of pattern occurrences. However, it is

clear that Algorithm 2 remains the best option when considering a huge data set where we typically have $\ell_r << \ell = \ell_1 + ... + \ell_r$.

Long sequences and low complexity pattern

Algorithm 3: Compute the $\mathcal{T}_{n+1}(G_N(y))$ in the case of a homogeneous model using power computations. The workspace complexity is $O(n \times K \times L^2)$ with $K = \log_2(\max\{\ell_1 - d, \ell_2 - \ell_1, ..., \ell_r - \ell_{r-1}\})$. The precomputation time complexity is $O(n^2 \times K \times L^3)$. All $\mathcal{T}_{n+1}(G_{N_{\ell_j}}(y))$ are computed with a total time complexity $O(r \times n^2 \times K \times L^3)$. The product updates of $U(y)$ result in an additional time complexity of $O(r \times n^2)$.

Require: The matrices $\mathbf{P}$ and $\mathbf{Q}$, for all $1 \le j \le r$, the starting distributions $\mathbf{m}_d^j$, the length ℓ_j (assuming $\ell_0 \overset{\text{def}}{=} d \leqslant \ell_1 \leqslant \ldots \leqslant \ell_r$), a $O(n \times L)$ workspace to keep the current values of $\mathbf{E}(y)$ (a dimension L polynomial row-vector of degree $n + 1$) and $U(y)$ (a polynomial of degree $n + 1$), and a $O(n \times K \times L^2)$ workspace to store the values of $\mathbf{M}_{2^k}(y)$ with $0 \le k \le K = \log_2(\max\{\ell_1 - d, \ell_2 - \ell_1, ..., \ell_r - \ell_{r-1}\})$.

// *Precompute all* $\mathbf{M}_{2^k}(y)$
$\mathbf{M}_{2^0}(y) \leftarrow \mathbf{P} + y\mathbf{Q}$
for $k = 1, ..., K$ **do**
$\mathbf{M}_{2^k}(y) \leftarrow \mathcal{T}_{n+1}(\mathbf{M}_{2^{k-1}}(y) \times \mathbf{M}_{2^{k-1}}(y))$
// *Initialization*
$U(y) \leftarrow 1$ and $\mathbf{E}(y) \leftarrow \mathbf{1}$
// *Loop on sequences*
for $j = 1, ..., r$ **do**
compute $\mathbf{M}_{\ell_j - \ell_{j-1}}(y)$ using a binary decomposition and set $\mathbf{E}(y) \leftarrow \mathcal{T}_{n+1}(\mathbf{M}_{\ell_j - \ell_{j-1}}(y)\mathbf{E}(y)^{\mathrm{T}})$
optionally return $\mathcal{T}_{n+1}(G_{N_{\ell_j}}(y)) = \mathbf{m}_d^j \mathbf{E}(y)^{\mathrm{T}}$
$U(y) \leftarrow \mathcal{T}_{n+1}(U(y) \times \mathbf{m}_d^j \mathbf{E}(y)^{\mathrm{T}})$
Output: return $\mathcal{T}_{n+1}(G_N(y)) = U(y)$

We now consider the case where ℓ_r is large (ex: ℓ_r = 100, 000 or 1, 000, 000 or more). With Algorithm 2, the time complexity is linear with ℓ_r,and may then result in an unacceptable running time. It is however possible to turn this into a logarithmic complexity by computing directly the powers of $(\mathbf{P} + y\mathbf{Q})$. This particular idea is not new in itself and has already been used in the context of pattern problems by several authors [50,51]. The novelty here is to apply this approach to a data set of multiple sequences.

If we denote by $\mathbf{M}_i(y) \overset{\text{def}}{=} \mathcal{T}_{n+1}((\mathbf{P} + y\mathbf{Q})^i)$, it is clear that all $\mathbf{M}_{2^k}(y)$ can be computed (and stored) for $0 \le k \le K$ with a space complexity $O(n \times K \times L^2)$ and a time complexity $O(n^2 \times K \times L^3)$. It is therefore possible to compute all $G_{N_{\ell_j}}(y)$ using the same approach as in Algorithm 2 except that all recursive updates of $\mathbf{E}(y)$ are replaced by direct power computations. This results in Algorithm 3 whose total complexities are $O(n \times K \times L^3)$ in space and $O(r \times n^2 \times K \times L^3)$ in time with $K = \log_2(\max\{\ell_1 - d, \ell_2 - \ell_1, ..., \ell_r - \ell_{r-1}\})$. The key feature of this algorithm is that we have replaced ℓ_r by the quantity K, which is typically dramatically smaller when we consider large ℓ_r. The drawback of this approach is that the space complexity is now quadratic with the pattern complexity L, and that the time complexity is cubic with L. As a consequence, it is not suitable to use Algorithm 3 for a pattern of high complexity.

Long sequences and high complexity pattern

If we now consider a moderate or high complexity pattern, we cannot accept either a cubic complexity with L or even a quadratic complexity. Hence only Algorithms 1 or 2 are appropriate. However, if we assume that our data set contains at least one long sequence, it may be difficult to perform the computations. This is why we introduce an approach that allows computing $G_N(y) = \mathbf{m}_d(\mathbf{P} + y\mathbf{Q})^{\ell - d}\mathbf{1}^{\mathrm{T}}$ for large ℓ and L. The technique is directly inspired from the partial recursion introduced in [51] to compute $g(y) = \mathbf{m}_d(\mathbf{P} + \mathbf{Q} + y\mathbf{Q})^{\ell - d}\mathbf{1}^{\mathrm{T}}$.

In this particular section, we assume that $\mathbf{P}$ is an irreducible and aperiodic matrix. We denote by λ the largest magnitude of the eigenvalues of $\mathbf{P}$, and by v the second largest magnitude of the eigenvalues of $\mathbf{P}/\lambda$. For all $i \ge 0$ we consider the polynomial vector $\mathbf{F}_i(y) \overset{\text{def}}{=} \left(\tilde{\mathbf{P}} + y\tilde{\mathbf{Q}}\right)^i \mathbf{1}^{\mathrm{T}}$, where $\tilde{\mathbf{P}} \overset{\text{def}}{=} \mathbf{P}/\lambda$ and $\tilde{\mathbf{Q}} \overset{\text{def}}{=} \mathbf{Q}/\lambda$, and hence we have $G_N(y) = \lambda^{\ell - d}\mathbf{m}_d\mathbf{F}_{\ell - d}(y)$.

Like in [51], the idea is then to recursively compute finite differences of $\mathbf{F}_i(y)$ up to the point where these differences asymptotically converge at a rate related to v^i. We then derive an approximated expression for $\mathbf{F}_{\ell - d}(y)$ using only terms such as $i \le \alpha$. Unfortunately, this approach through partial recursion suffers the same numerical instabilities as in [51] when computations are performed in floating point arithmetic. For this reason, we chose here not to go further in that direction until a more extensive study has been conducted.

Results and discussion

Comparison with known algorithms

To the best of our knowledge, there is no record of any method that allows computing the distribution of a random pattern count in a set of heterogeneous Markov sequences. However, a great number of concurrent approaches exists to perform the computations for a single sequence, where the result for a set of sequences is obtained by convolutions.

For the heterogeneous case for a single sequence of length ℓ, any kind of Markov chain embedding techniques [48,52] may be used to get the expression of one $G_{N_\ell}(y)$ up to degree $n + 1$ with complexity $O(\ell \times n \times |\mathcal{A}| \times L)$. In this respect, there is little novelty in Algorithm 1, except that it allows avoiding the $O(r \times n^2)$ additional cost of the convolution product, which could be a great advantage. In the homogeneous case, the main interest of our approach is its ability to exploit the repeated nature of the data (a set of sequences) to save

computational time. This is typically what it is done in Algorithm 2.

From now on, we will only consider the problem of computing the exact distribution of the pattern count N_ℓ in a single (long) sequence of length ℓ generated by a homogeneous Markov source, and compare the novel approaches introduced in this paper to the most efficient methods available.

One of the most popular of these methods consists in considering the bivariate moment generating function

$$G(y,z) \stackrel{\text{def}}{=} \sum_{n\geqslant 0,\ell\geqslant d} \mathbb{P}(N_\ell = n) y^n z^\ell \tag{11}$$

where y and z are dummy variables. Thanks to Equation (6) it is easy to show that

$$G(y,z) = z^d \times \mathbf{m}_d(\mathbf{Id} - z(\mathbf{P} + y\mathbf{Q}))^{-1}\mathbf{1}^{\mathrm{T}} \tag{12}$$

It is thus possible to extract the coefficients from $G(y, z)$ using fast Taylor expansions. This interesting approach has been suggested by several authors including [46] or [48] and is often referred to as the "golden" approach for pattern problems. However, in order to apply this method, one should first use a Computer Algebra System (CAS) to perform the bivariate polynomial resolution of the linear system $(\mathbf{Id} - z(\mathbf{P} + y\mathbf{Q}))\,\mathbf{x}^{\mathrm{T}} = \mathbf{1}^{\mathrm{T}}$. This may result in a complexity in $O(L^3)$ which is not suitable for high complexity patterns. Alternatively, one may rely on efficient linear algebra methods to solve sparse systems like the sparse LU decomposition. But the availability of such sophisticated approaches, especially when working with bivariate polynomials, is likely to be an issue.

Once the bivariate rational expression of $G(y,z)$ is obtained, performing the Taylor expansions still requires a great deal of effort. This usually consists in first performing an expansion in z in order to get the moment generating function $G_{N_\ell}(y)$ of N_ℓ for a particular length ℓ. The usual complexity for such task is $O(D_z^3 \times \log \ell)$ where D_z is the denominator degree (in z) in $G(y, z)$. In this case however, there is an additional cost due to the fact that these expansions have to be performed with polynomial (in y) coefficients. Finally, a second expansion (in y) is necessary to compute the desired distribution. Fortunately, this second expansion is done with constant coefficients. It nevertheless results in a complexity $O(D_y^3 \times \log n)$ where D_y is the degree of the denominator in $G_{N_\ell}(y)$ and n the observed number of occurrences.

In comparison, the direct computation of $G_{N_\ell}(y) = \mathbf{m}_d(\mathbf{P} + y\mathbf{Q})\mathbf{1}^{\mathrm{T}}$ by binary decomposition (Algorithm 2) is much simpler to implement (relying only on floating point arithmetics) and is likely to be much more effective in practice.

Recently, [50] suggested to compute the full bulk of the exact distribution of N_ℓ through Equation (6) using a power method like in Algorithm 3, with the noticeable difference that all polynomial products are performed using Fast Fourier Transforms (FFT). Using this approach, and a very careful implementation, one can compute the full distribution with a complexity $O(L^3 \times \log_2 \ell \times n_{\max} \log_2 n_{\max})$ where $n_{\max}$ is the maximum number of pattern occurrences in the sequence, which is better than Algorithm 3. There is however a critical drawback to using FFT polynomial products: the resulting coefficients are only known with an absolute precision equal to the largest one times the relative precision of floating point computations. As a consequence, the distribution is accurately computed in its center region, but not in its tails. Unfortunately, this is precisely the part of the distribution that matters for significant P-values, which are obviously the number one interest in pattern study. Finally, let us remark that the approach introduced by [50] is only suitable for low or moderate complexity patterns.

The new algorithms we introduce in this paper have the unique feature to be able to deal with a set of heterogeneous sequences. These algorithms, compared to the ones found in the literature, also display similar or better complexities. Last but not least, the approaches we introduce here only rely on simple linear algebra and are hence far easier to implement than their classical alternatives.

Illustrative examples

In this part we consider several examples. We start with a simple toy-example for the purpose of illustrating the techniques, and we then consider three real biological applications.

A toy-example

In this part we give a simple example to illustrate the techniques and algorithms presented above. We consider the pattern $\mathcal{W}$ = {abab, abaab, abbab} over the binary alphabet $\mathcal{A}$ = {a, b}. The minimal DFA that recognizes the language $\mathcal{L} = \mathcal{A}^*\mathcal{W}$ (which is the set of all texts over $\mathcal{A}$ ending with occurrence of $\mathcal{W}$) is then given in Figure 1.

Let us now consider the following set of $r = 3$ sequences:

$$x^1 = \text{abaabbaba}\ (\ell_1 = 9),\quad x^2 = \text{bababb}\ (\ell_2 = 6)\quad \text{and}\quad x^3 = \text{abbaabab}\ (\ell_3 = 8).$$

We process these sequences to the DFA of Figure 1 (starting each sequence in the initial state 0) to get the observed state sequences $\tilde{x}^1$, $\tilde{x}^2$ and $\tilde{x}^3$:

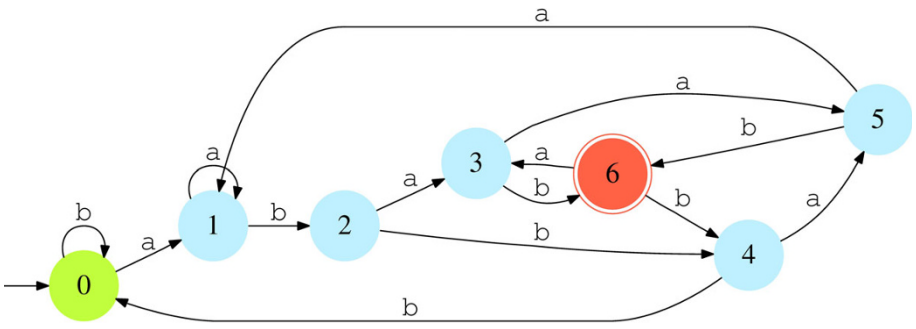

Figure 1 Minimal DFA that recognizes the language $\mathcal{L}$ = {a, b}* $\mathcal{W}$ with $\mathcal{W}$ = {abab, abaab, abbab}.

pos.	–	1	2	3	4	5	6	7	8	9
x^1	–	a	b	a	a	b	b	a	b	a,
$\tilde{x}^1$	0	1	2	3	5	**6**	4	5	**6**	3

pos.	–	1	2	3	4	5	6
x^2	–	b	a	b	a	**b**	b
$\tilde{x}^2$	0	0	1	2	3	**6**	4

and

pos.	–	1	2	3	4	5	6	7	8
x^3	–	a	b	b	a	a	b	a	b.
$\tilde{x}^3$	0	1	2	4	5	1	2	3	**6**

Therefore, Sequence x^1 contains $n_1 = 2$ occurrences of the pattern (ending in positions 5 and 8), Sequence x^2 contains $n_2 = 1$ occurrence (ending in position 5) and Sequence x^3 contains $n_3 = 1$ occurrence (ending in position 8).

Let us now consider X^1, X^2 and X^3, three homogeneous order $d = 1$ Markov chains of respective lengths ℓ_1, ℓ_2 and ℓ_3 such that X^1 and X^3 start with a, and X^2 starts with b, and the transition matrix of which is given by:

$$\pi = \begin{pmatrix} 0.7 & 0.3 \\ 0.4 & 0.6 \end{pmatrix}.$$

The corresponding state sequences $\tilde{X}^1$, $\tilde{X}^2$ and $\tilde{X}^3$ are hence order 1 homogeneous Markov chains defined over $\mathcal{Q}'$ = {0, 1,2, 3, 4, 5, 6} with the starting distributions $\mathbf{m}_1^1 = \mathbf{m}_1^3$ = (0 1 0 0 0 0 0), $\mathbf{m}_1^2$ = (1 0 0 0 0 0 0) (since starting from 0 in the DFA of Figure 1, a leads to state 1 and b to state 0) and with the following transition matrix (please note that transitions belonging to **Q** are marked with a '*'. The others ones belong to **P**):

$$\mathbf{T} = \begin{pmatrix} 0.6 & 0.4 & – & – & – & – & – \\ – & 0.7 & 0.3 & – & – & – & – \\ – & – & – & 0.4 & 0.6 & – & – \\ – & – & – & – & – & 0.3 & 0.7\,* \\ 0.6 & – & – & – & – & 0.4 & – \\ – & 0.7 & – & – & – & – & 0.3\,* \\ – & – & – & 0.4 & 0.6 & – & – \end{pmatrix}$$

A direct application of Corollary 3 therefore gives $G_{N_1}(y) = 0.743104 + 0.208944y + 0.0450490y^2 + 0.0029030y^3$ for the moment generating function of N_1 (the number of pattern occurrences in X^1);

$G_{N_2}(y) = 0.94816 + 0.05184y$ for the moment generating function of N_2 (the number of pattern occurrences in X^2); and $G_{N_3}(y) = 0.7761376 + 0.1880064y + 0.0353376y^2 + 0.0005184y^3$ for the moment generating function of N_3 (the number of pattern occurrences in X^3). One should note that occurrences of $\mathcal{W}$ are strongly disfavored in Sequence X^2 since it starts with b. We then derive from these expressions the value of the moment generating function $G_N(y)$ of $N = N_1 + N_2 + N_3$:

$$G_N(y) = G_{N_1}(y) \times G_{N_2}(y) \times G_{N_3}(y) = 0.5468522 + 0.3161270y + 0.1109456y^2 + 0.0227431y^3 + 0.0030882y^4 + 0.0002358y^5 + 0.0000080y^6 + 7.801\times10^{-8}y^7 \quad (13)$$

Since we observe a total of $n = n_1 + n_2 + n_3 = 4$ occurrences of Pattern $\mathcal{W}$, the P-value of over-representation is given by

$$\begin{aligned}\mathbb{P}(N \geqslant 4) &= \mathbb{P}(N = 4) + \mathbb{P}(N = 5) + \mathbb{P}(N = 6) + \mathbb{P}(N = 7) \\ &= 0.0030882 + 0.0002358 + 0.0000080 + 7.801\times10^{-8} \\ &= 3.33\times10^{-3}\end{aligned} \quad (14)$$

Let us finally compare the exact distribution of N', the number of pattern occurrences over $X = X_1 \ldots X_\ell$ with $\ell = \ell_1 + \ell_2 + \ell_3 - 2 \times$ offset, and a homogeneous order 1 Markov chain with transition matrix π:

offset	0	1	2	3	4	5	6
$10^2 \times \mathbb{P}(N' \geqslant 4 \mid X_1 = a)$	2.252	1.647	1.158	0.743	0.447	0.249	0.043
$10^2 \times \mathbb{P}(N' \geqslant 4 \mid X_1 = b)$	1.561	1.088	0.706	0.417	0.223	0.064	0.002

As $\mathcal{W}$ contains both words of lengths 4 and 5, offset should be set either to 3 or 4. However, for both these values, $10^2 \times \mathbb{P}(N' \geq 4)$ (either when X_1 = a or when X_1 = b) differs from the reference exact P-value $10^2 \times \mathbb{P}(N \geq 4) = 0.333$.

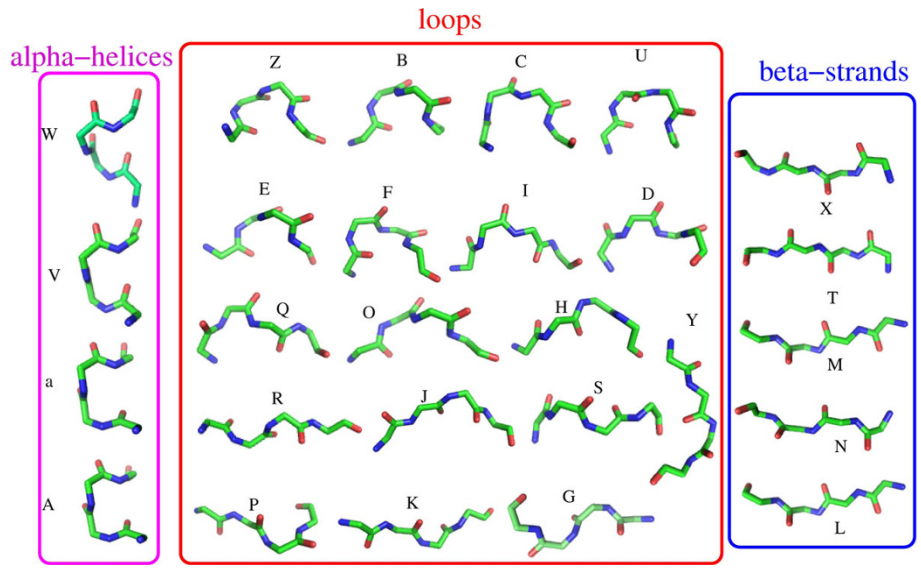

Figure 2 Geometry of the 27 structural letters of the HMM-27 structural alphabet.

Structural motifs in protein loops

Protein structures are classically described in terms of secondary structures: α-helices, β-strands and loops. Structural alphabets are an innovative tool that allows describing any three-dimensional (3D) structure by a succession of prototype structural fragments. We here use HMM-27, an alphabet composed of 27 structural letters (it consists in a set of average protein fragments of four residues, called structural letters, which is used to approximate the local backbone of protein structures through a HMM): 4 correspond to the alpha-helices, 5 to the beta-strands and the 18 remaining ones to the loops (see Figure 2) [53]. Each 3D structure of ℓ residues is encoded into a linear sequence of HMM-27 structural letters and results in a sequence of ℓ - 3 structural letters since each overlapping fragment of four consecutive residues corresponds to one structural letter.

We consider a set of 3D structures of proteins presenting less than 80% identity and convert them into sequences of structural letters. Like in [54], we then make the choice to focus only on the loop structures which are known to be the most variable ones, and hence the more challenging to study. The resulting loop structure data set is made of 78,799 sequences with length ranging from 4 to 127 structural letters.

In order to study the interest of the single sequence approximation described in the "Single sequence approximation" section, we first perform a simple experiment. We fit an order 1 homogeneous Markov model on the original data set, and then simulate a random data set with the same characteristics (loop lengths and starting structural letters). We then compute the z-score - these quantities are far easier to compute than the exact P-values and they are known to perform well for pattern problems as long as we consider events in the center of the distribution, and such events are precisely the ones expected to occur with a simulated data set - of the 77, 068 structural words of size 4 that we observe in the data, using simulated data sets under the single sequence approximation. We observe that high z-scores are strongly over-represented in the simulated data set: for example, we observed 264 z-scores of magnitude greater than 4, which is much larger than the expected number of 4.88 under H_0. This observation clearly demonstrates that the single sequence approximation completely fails to capture the distribution of structural motifs in this data set. Indeed this experiment initially motivated the present work by putting emphasis on the need for taking into account fragmented structure of the data set.

We further investigate the edge effects in the data set by comparing the exact P-values obtained under the single sequence approximation. Table 1 gives the results for a selected set of 14 motifs whose occurrences range from 4 to 282. We can see that the single sequence approximation with an offset of 0 clearly differs from the exact value: e. g., Pattern ODZR has an exact P-value of 5.78×10^{-5} and an approximate one of 2.81×10^{-4}; Pattern BZOU has an exact P-value of 2.56×10^{-11} and an approximate one of 4.49×10^{-5}.

As explained in the Methods section, these differences may be caused by the overlapping positions in the artificial single sequence where the pattern cannot occur in the fragmented data set. Since we consider patterns of size 4, a canonical choice of offset is 4 - 1 = 3. We can

Table 1 P-values for structural patterns in protein loop structures using exact computations or the single sequence approximation (SSA) with offset or not.

Structural pattern	*n*	Exact	SSA (no offset)	SSA (offset = 3)
KYNH	16	1.62×10^{-2}	5.95×10^{-1}	8.43×10^{-2}
PNKK	7	2.20×10^{-2}	6.68×10^{-2}	9.19×10^{-3}
JLPQ	25	1.37×10^{-3}	4.89×10^{-1}	2.19×10^{-2}
QYHB	110	1.71×10^{-3}	9.46×10^{-1}	2.59×10^{-3}
ODZR	4	5.78×10^{-5}	2.81×10^{-4}	5.49×10^{-5}
CPBQ	27	5.69×10^{-6}	3.07×10^{-3}	3.81×10^{-6}
ZGBZ	50	3.45×10^{-7}	4.84×10^{-2}	9.71×10^{-6}
BZOU	40	2.56×10^{-11}	4.49×10^{-5}	1.22×10^{-9}
UOEI	52	5.74×10^{-16}	1.96×10^{-10}	2.30×10^{-17}
EGZD	58	3.19×10^{-32}	1.91×10^{-23}	1.26×10^{-32}
GIYC	149	1.05×10^{-41}	1.06×10^{-30}	3.85×10^{-51}
DRPI	282	7.26×10^{-167}	9.08×10^{-174}	3.56×10^{-222}

see in Table 1 the effects of this correction. For most patterns, this approach improves the reliability of the approximations, even if we still see noticeable differences. For instance we get an approximated P-value larger than the exact one for Pattern BZOU, and an approximated P-value smaller than the exact one for Pattern UOEI. For other patterns, this correction is ineffective and gives even worse results than with an offset of 0. For example, Pattern DRPI has an exact P-value of 7.26×10^{-167} and an approximate P-value with an offset of 3 equal to 3.56×10^{-222}, while the approximation with no offset gives a P-value of 9.08×10^{-174}.

Hence it is clear that the forbidden overlapping positions alone cannot explain the differences between the exact results and the single sequence approximation. Indeed, there is another source of edge effects which is connected to the background model. Since each sequence of the data set starts with a particular letter, the marginal distribution differs from the stationary one for a number of positions that depends on the spectral properties of the transition matrix. It is well known that the magnitude μ of the second eigenvalue of the transition matrix plays here a key role since the absolute difference between the marginal distribution at position i and the stationary distribution is $O(\mu^i)$. In our example, $\mu = 0.33$, which is very large, leads to a slow convergence toward the stationary distribution: we need at least 30 positions to observe a difference below machine precision between the two distributions. Such an effect is usually negligible for long sequences where $30 << \ell$, but is critical when considering a data set of multiple short sequences.

However, this effect might be attenuated on the average if the distribution of the first letter in the data set is close to the stationary distribution. Figure 3 compares these two distributions. Unfortunately in the case of structural letters, there is a drastic difference between these distributions.

The example of structural motifs in protein loop structures illustrates the importance of explicitly taking into account the exact characteristics of the data set (number and lengths of sequences) when the single sequence approximation appears to be completely unreliable. As explained above, this may be due both to the great differences between the starting and the stationary distributions, as well as to a slow convergence and to the problem of forbidden positions.

PROSITE signatures in protein sequences

We consider the release 20.44 of PROSITE (03-Mar-2009) which encompasses 1, 313 different patterns described by regular expressions of various complexity [9]. PROSITE currently contains patterns and specific profiles for more than a thousand protein families or domains. Each of these signatures comes with documentation providing background information on the structure and function of these proteins. The shortest regular expression is for pattern PS00016: RGD, i. e., an exact succession of arginine, glycine and aspartate residues. This pattern is involved in cell adhesion. The longest regular expression, on the opposite, is for pattern PS00041:

[KRQ][LIVMA].(2)[GSTALIV]FYWPGDN.(2)[LIVMSA].(4, 9)[LIVMF].{PLH}[LIVMSTA][GSTACIL]GPKF.[GANQRF][LIVMFY].(4, 5)[LFY].(3)[FYIVA]{FYWHCM}{PGVI}.(2)[GSADENQKR].[NSTAPKL][PARL] (note that X means "any aminoacid", brackets denote a set of possible letters, braces a set of forbidden letters, and parentheses repetitions -fixed number of times or on a given range). This is the signature of the DNA-binding domain of the araC family of bacterial regulatory proteins.

This data set is useful to explore one of the key points of our optimal Markov chain embedding method using

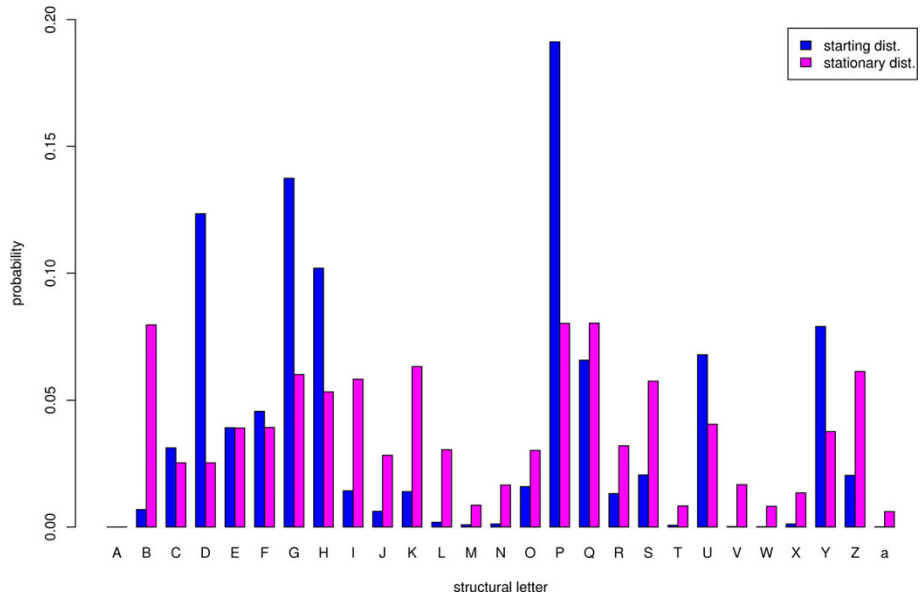

Figure 3 Starting and stationary distributions of the 27 structural letters in the loop structure data set.

DFAs: the impact of the pattern complexity L. For this purpose, we first build 1-unambiguous (since we want to work with an order 1 Markov model) associated DFAs for 1,276 PROSITE patterns (37 patterns requiring a prohibiting computation time and/or memory were not computed). The repartition of the resulting pattern complexities is shown in Figure 4. There is a peak in the distribution at 2, meaning that many DFAs have ≃ 100 states. The smallest DFA is obtained for the `RGD` pattern (22 states), and the largest is for APPLE (`PS00495`) which is represented by the regular expression `C.(3)[LIVMFY].(5)[LIVMFY].(3)[DENQ][LIVMFY].(10)C.(3)CT.(4)C.[LIVMFY]F.[FY].(13,14)C.[LIVMFY][RK].[ST].(14,15)SG.[ST][LIVMFY].(2)C` which has 837, 507 states. The mean computing time of the DFA is 3 minutes, but 50% of the DFA could be computed in less than 0.01s, and 95% in less than 9s.

In Table 2, we can see that if short regular expressions usually lead to low complexity patterns, it is difficult to predict the result for longer regular expressions. For instance, the PROSITE signatures PUR_PYR_PR_-TRANSFER and ADH_ZINC have the same size, but the former has a complexity of $L = 102$ while the latter has a complexity of $L = 478$. Indeed, we know from the theory of language and automata [55] that the minimal DFA corresponding to a regular expression of size R has a size L verifying $L \leq 2^R$. Fortunately, in practice, L is usually dramatically smaller than this upper bound.

We now consider the complete proteome of the bacteria *Escherichia coli* (File NC_000913.faa, retrieved at ftp://ftp.ncbi.nih.gov/genomes/Bacteria/Escherichia_coli_K_12_substr__MG1655/). This data set encompasses a total of 4, 131 protein sequences with lengths ranging from 14 to 2, 358 aminoacids. We fit on this data set a homogeneous order 1 Markov model which is used to derive over-representation P-values of patterns.

Like for structural letters, we compare the exact P-values to the ones obtained using the single sequence approximation, see Table 3. Unlike in Table 1, we see here that the single sequence approximation performs already well with no offset, but that the use of the appropriate offset further improves this approximation.

This result is surprising, since, in this case, the starting distribution of the model strongly differs from the stationary distribution. Indeed, it is a biological fact that all protein sequences start with a methionine (`M`). As a consequence, it is hence clear that the starting distribution and the stationary distribution of the model strongly differ. This observation obviously does not favor the single sequence approximation. But in this example, this effect is corrected by the rapid convergence of the marginal distribution toward the stationary distribution ensured by a very low second magnitude eigenvalue of the matrix: $\mu = 0.049$. We expect the same kind of behavior for the high complexity patterns of Table 4 but because of the numerical instabilities in the partial recursion approach suggested in the "Long sequences and high complexity pattern" section, unfortunately it was impossible to perform the computations for the single sequence approximation for such pattern in a reasonable time. However, it is possible to perform

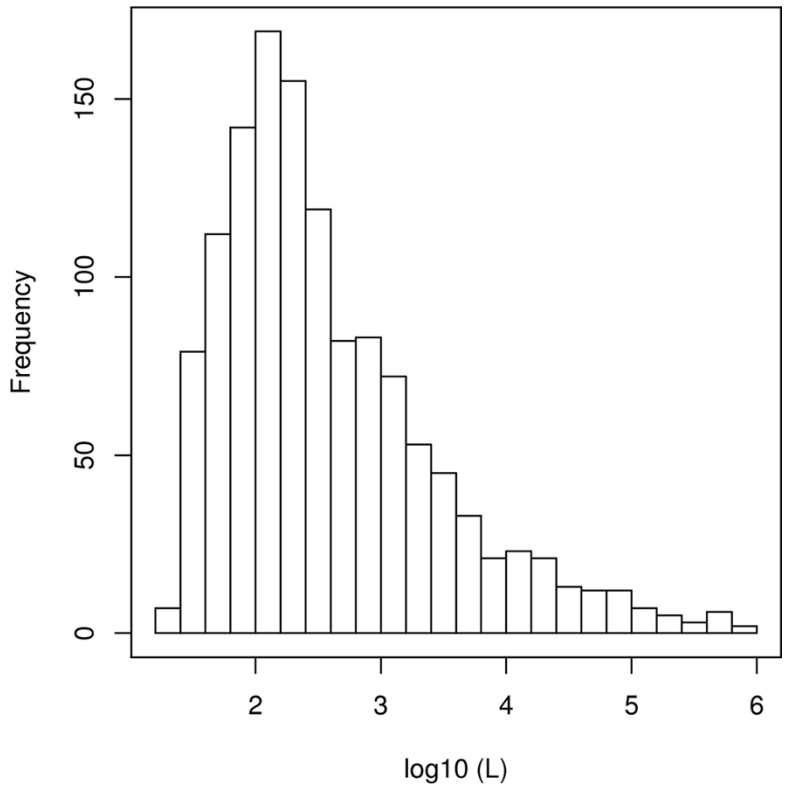

Figure 4 Histogram of the $\log_{10}(L)$ for 1, 276 PROSITE patterns in the framework of an order 1 Markov model. Note that the 0.1% patterns with the largest complexities have been removed from the graph in order to improve readability.

Table 2 Size of the regular expression (regex) and pattern complexity (*L*) for a selected subset of PROSITE signatures.

PROSITE signature	Accession number	pattern size	*L*
RGD	PS00016	3	22
ER_TARGET	PS00014	3	28
PPASE	PS00387	7	41
ALDEHYDE_DEHYDR_GLU	PS00687	8	44
PROKAR_NTER_METHYL	PS00409	21	46
GLY_RADICAL_1	PS00850	9	77
PEP_ENZYMES_PHOS_SITE	PS00370	12	96
PUR_PYR_PR_TRANSFER	PS00103	13	102
PILI_CHAPERONE	PS00635	18	226
SIGMA54_INTERACT_2	PS00676	16	313
EFACTOR_GTP	PS00301	16	320
ALDEHYDE_DEHYDR_CYS	PS00070	12	331
ADH_ZINC	PS00059	13	478
THIOLASE_1	PS00098	19	637
SUGAR_TRANSPORT_1	PS00216	15 to 17	796
FGGY_KINASES_2	PS00445	21 to 22	2668
PTS_EIIA_TYPE_2_HIS	PS00372	16	2758
MOLYBDOPTERIN_PROK_3	PS00551	27 to 28	3907
SUGAR_TRANSPORT_2	PS00217	26	6889

Table 3 P-values for a selection of PROSITE patterns of low (or moderate) complexities using the complete proteome of *Escherichia coli* (NC_000913.faa).

PROSITE signature	*n*	Exact	SSA with no offset	SSA (offset)
RGD	215	5.35×10^{-1}	5.91×10^{-1}	5.55×10^{-1}(2)
ER_TARGET	72	4.01×10^{-2}	5.21×10^{-2}	4.70×10^{-2}(2)
PPASE	3	2.60×10^{-2}	2.76×10^{-2}	2.63×10^{-2}(6)
ALDEHYDE_DEHYDR_GLU	12	1.99×10^{-5}	2.41×10^{-5}	1.95×10^{-5}(7)
PROKAR_NTER_METHYL	10	6.79×10^{-3}	8.01×10^{-3}	5.10×10^{-3}(20)
GLY_RADICAL_1	6	1.58×10^{-6}	1.86×10^{-6}	1.60×10^{-6}(8)
PEP_ENZYMES_PHOS_SITE	4	1.49×10^{-10}	1.74×10^{-10}	1.49×10^{-10}(12)
PUR_PYR_PR_TRANSFER	7	2.15×10^{-14}	2.75×10^{-14}	2.10×10^{-14}(12)

the exact computation for these high complexity patterns using Algorithm 2.

Considering the multi-testing problem of this study (we consider a total of 1, 276 PROSITE signatures), we can set a significance threshold of 7.84×10^{-7} at level 0.1% using a Bonferonni correction. Even at this stringent level, it is clear that many of the considered PROSITE signatures (2 out of 8 in Table 3, and 9 out of 11 in Table 4) are over-represented compared to our homogeneous order 1 Markov background model. However, this result is not a surprise since these patterns actually correspond to very precise functional signatures which are therefore expected to be strongly maintained through evolution in order keep their functional activities.

DNA motifs in gene upstream regions

Transcription factors regulate the expression of genes by activating or repressing the RNA polymerase. This is done by specific binding of the transcription factors (TFs) onto DNA, in proximity to the target genes, usually in the upstream regions. The transcription binding signatures on DNA are thus biologically important patterns.

Table 4 Exact P-values for a selection of PROSITE patterns of high complexities using the complete proteome of *Escherichia coli* (NC_000913.faa). We use an order 1 homogeneous Markov model estimated over the data set.

PROSITE signature	*n*	Exact
PILI_CHAPERONE	10	3.27×10^{-46}
SIGMA54_INTERACT × 2	12	1.58×10^{-42}
EFACTOR_GTP	8	4.43×10^{-20}
ALDEHYDE_DEHYDR_CYS	11	5.63×10^{-9}
ADH_ZINC	12	8.93×10^{-16}
THIOLASE_1	5	5.76×10^{-9}
SUGAR_TRANSPORT_1	18	3.75×10^{-8}
FGGY_KINASES_2	5	2.14×10^{-4}
PTS_EIIA_TYPE_2_HIS	8	7.19×10^{-19}
MOLYBDOPTERIN_PROK_3	11	2.59×10^{-35}
SUGAR_TRANSPORT_2	10	1.22×10^{-5}

We retrieved the sequence of transcription factor binding sites of *Saccharomyces cerevisiae* on the YEASTRACT website http://www.yeastract.com/consensuslist.php and searched for a subset of these transcription factor binding sites in the upstream regions of yeast genes, retrieved on the Regulatory Sequence Analysis Tools website [56]http://rsat.ulb.ac.be/rsat/. This data set comprises a total of 1,371 upstream sequences between positions -800 and -1 (the length is hence $\ell = 800$ for each sequence).

On these data, we first fit an order 1 homogeneous Markov model. Since there is little difference between the starting distribution observed in the data set over $\mathcal{A} = \{\mathtt{A}, \mathtt{C}, \mathtt{G}, \mathtt{T}\}$(0.30 0.16 0.19 0.35) and the stationary distribution (0.32 0.18 0.18 0.32), and since the magnitude of the second eigenvalue of the transition matrix is fairly low ($\mu = 0.092$), we do not expect a great difference between the exact computations and the single sequence approximation. However, since exact computations are easily tractable, we do not further consider the single sequence approach for this particular problem.

We can see in Table 5 the P-values (column "homogeneous") of a selection of known TFs (marked with a star) as well as arbitrary candidate patterns. Several known TFs appear to be highly significant (e.g., TF AAGAAAAA with a P-value of 1.31×10^{-99}) while others are not (e.g., TF WWWTTTGCTCR with a P-value of 4.15×10^{-1}). It is the same for arbitrary candidate patterns. These results are difficult to interpret since these variations may be due either to statistical problems (e.g., insufficient Markov order) or real functional activities. Moreover, it is obviously impossible to distinguish a significant pattern which is a real TF of the organism from a significant pattern which is directly or indirectly implicated in another biochemical process.

We now want to get rid of the homogeneous assumption of the model in an attempt to get a better fitting on the data. A simple way to achieve this is to perform a point-wise estimation of our transition function at position i by fitting the model on a window of

Table 5 P-values for several DNA patterns (known transcription factors are marked with a star) in the upstream region data set.

DNA pattern	*n*	*L*	homogeneous	heterogeneous
CGCACCC*	28	10	2.95×10^{-3}	3.74×10^{-3}
AAGAAAAA*	427	11	1.31×10^{-99}	1.29×10^{-99}
AACAACAAC	25	10	1.76×10^{-6}	1.38×10^{-6}
TCCGTGGA*	22	11	1.12×10^{-6}	1.55×10^{-6}
GCGCGCGC	18	11	6.52×10^{-10}	1.65×10^{-9}
RTAAAYAA*	391	14	7.70×10^{-12}	1.68×10^{-12}
WWWTTTGCTCR*	15	17	4.15×10^{-1}	4.09×10^{-1}
AAAAAAAAAAAAAAAAAAAAAAAA	42	27	2.05×10^{-23}	2.14×10^{-22}
TAWWWWTAGM*	212	36	3.08×10^{-9}	3.04×10^{-9}
YCCNYTNRRCCGN*	11	40	3.10×10^{-2}	3.05×10^{-2}
GCGCNNNNNNGCGC	1	106	8.97×10^{-1}	8.84×10^{-1}
CGGNNNNNNNNCGG*	102	183	1.26×10^{-14}	1.73×10^{-13}
GCGCNNNNNNNNNGCGC	6	464	2.88×10^{-2}	2.84×10^{-2}

size w centered around i. Small values of w lead to better fitting, while large values lead to better smoothing (resulting in a homogeneous model if $w \geq \ell$, the length of the sequence). In this example, we achieve a satisfactory trade-off between the two extremes with an arbitrary choice of $w = 200$. We can see in Figure 5, that the model gives a unique profile for each transition probability (e.g., π_i(A, G) or π_i(G, G)), and these profiles are both quantitatively and qualitatively different from each other. In Figure 6 we consider the model in a more global way with the marginal distributions of the four nucleotides. According to this graph, it is clear that the upstream region has a bias in GC content that depends on the position. In particular, we observe a smaller GC content in the region [-200,

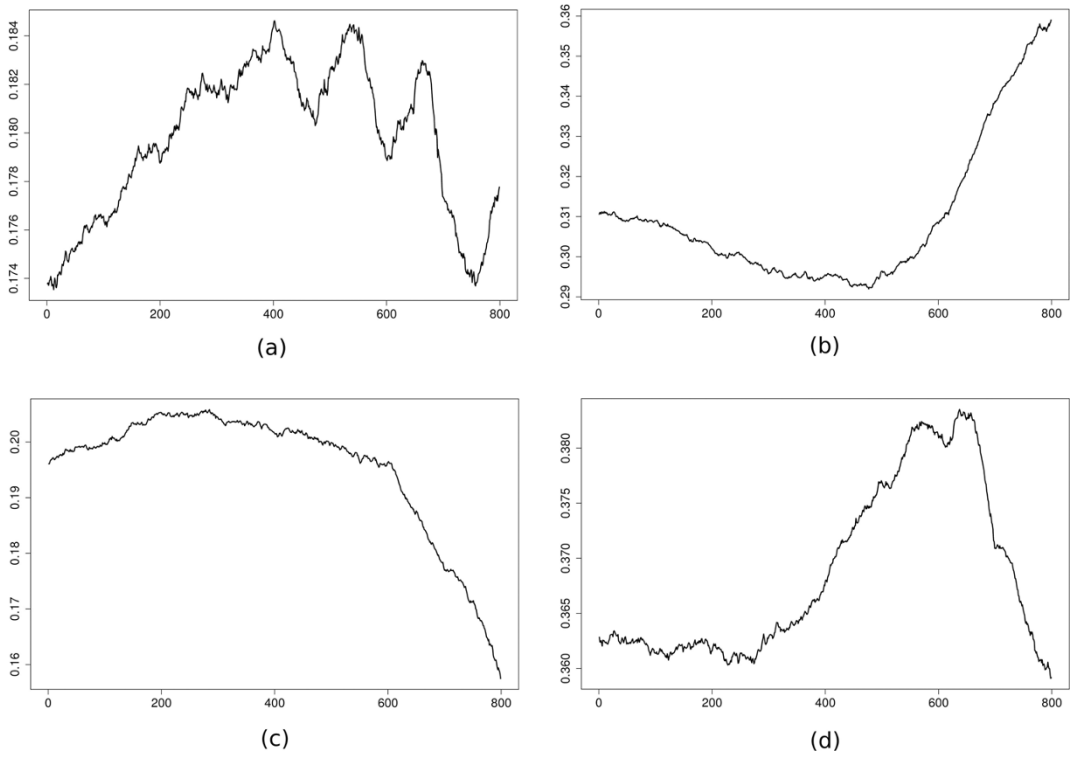

Figure 5 Some transitions of the order 1 heterogeneous Markov model fitted using a sliding window of size 200 on the upstream region data set. The plots respectively correspond to the following quantities: a) π_i(A, G); b) π_i(G, A); c) π_i(G, G); d) π_i(T, T), $1 \leq i \leq 800$.

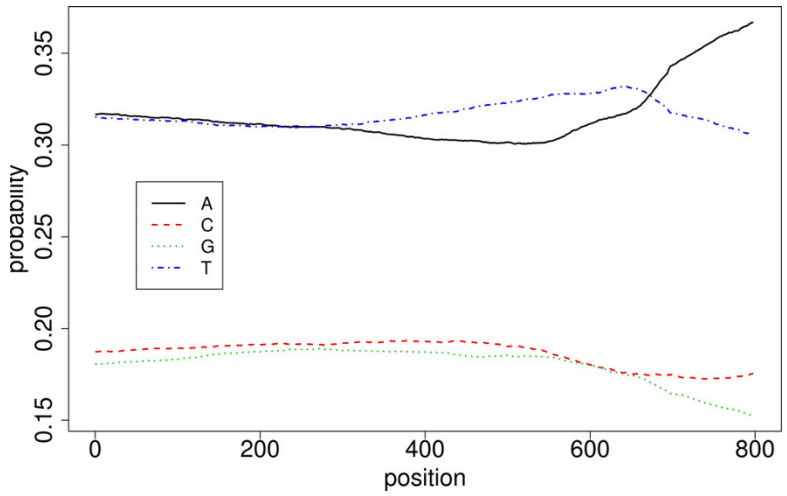

Figure 6 Marginal distribution of the four nucleotides along the 800 positions of a upstream region. The underlying model is an order 1 heterogeneous Markov model fitted using a sliding window of size 200 on the upstream region data set.

-1] (positions 601 to 800) than in the region [-800, -201] (positions 1 to 599).

Thanks to Algorithm 1, it is possible to compute the P-values of DNA patterns in our heterogeneous model. The results are given in Table 5 (column "heterogeneous"). For most patterns, we can see that the P-values obtained with this heterogeneous model are in fact very close to the ones obtained with the homogeneous one. There are however several patterns for which a ratio factor of 10 may appear between these two P-values (e.g., Pattern GCGCGCGC or CGGNNNNNNNNCGG).

Conclusion

In this paper, we introduce efficient algorithms to compute the exact distribution of random pattern counts in a set of multi-state sequences generated by a Markov source. These algorithms are able to deal both with low or high complexity patterns, and with either homogeneous or heterogenous Markov models.

This work, based on the recent notion of optimal Markov chain embedding through DFAs [46-49], is a natural extension of the methods and algorithms developed in [51] to obtain the first k^{th} moment of a random pattern count in one sequence. These computations of moments for a single sequence can easily be extended to a set of independent sequences by taking advantage of the fact that the cumulants (the first two cumulants are the expectation and the variance) of a sum of independent variables are the sum of the individual cumulants.

To the best of our knowledge, there currently exists no method specifically designed to compute the distribution of a random pattern count in a set of Markov sequences. However it exists a great deal of concurrent approaches to perform the computations for a single sequence, the result for a set of sequences being then obtained by convolution products. In this regard, Algorithm 1 has the interesting feature to completely avoid these convolutions and their possibly prohibitive $O(r \times n^2)$ additional cost (r being the number of sequences in the data set, and n being the observed number of occurrences), especially for large n. Algorithm 1 also has the advantage to be able to deal both with heterogenous models and high complexity patterns. However, with a complexity in $O(\ell \times n \times |\mathcal{A}| \times L)$ ($\ell = \ell_1 + ...+ \ell_r$ being the total length of the data set, s being the alphabet size, and L being the pattern complexity), this algorithm may be too slow when considering large data sets.

In the homogeneous model, Algorithm 2 can dramatically reduce the overall complexity by replacing ℓ by ℓ_r the length of the longest sequence in the data set. Moreover this algorithm can deal with high complexity patterns, but this requires performing convolution products. However, it is clear that Algorithm 2 remains the best option when considering a data set with a large number of sequences with reasonable length: $\ell_r << \ell = \ell_1 + ... + \ell_r$.

In the particular case where ℓ_r is too high (e.g., $\ell_r = 10^6$ or more), it may be necessary to switch from linear to logarithmic complexity. This may be achieved by several methods. When dealing with low complexity patterns, the best known approach consists in computing the bivariate rational moment generating function $G(y, z)$ of N_ℓ the random number of pattern occurrences in a random sequence of length ℓ and then to perform fast Taylor expansions (logarithmic complexity) to get the probabilities of interest. However, this approach requires sophisticated computation in bivariate polynomial

algebra, and has at least a cubic complexity with the denominator degree of the rational function $G(y, z)$ whose value may be too high to perform the computations. Alternatively, the power approach proposed in Algorithm 3 also achieves logarithmic complexity, but with an easier implementation relying only on basic floating point linear algebra.

For high complexity patterns, the cubic complexity in L is prohibitive and prevents using neither power computations nor the plain formal inversion that is required to compute $G(y, z)$. The partial recursion approach we introduce to deal with such a case appears to be a very interesting alternative, but its numerical instabilities in floating point arithmetic need to be further investigated. It is also possible to compute $G(y, z)$ by solving the corresponding sparse linear system with appropriate sparse linear algebra methods (e.g., sparse LU), but the availability of such methods for multivariate polynomial matrices is likely to be an issue. Moreover, one should expect the denominator degree of the moment generating function to increase with the pattern complexity which could thus result again in untractable computations.

Another tempting option is to ignore the particular structure of the data set by approximating the distribution of $N = N_1 + ... + N_r$ by the one of N', the random pattern count in a single sequence of length $\ell = \ell_1 + ... + \ell_r$. When one wants to use exact computations to get the distribution of N', the resulting complexity is likely to be far greater that the one required to obtain the exact distribution of N. However, these approximations might be interesting if the distribution of N' is obtained through efficient asymptotic approximations like Poisson or Large Deviations approximations. Unfortunately, we have seen in our applications that this approach is subject to important edge effects, especially when the convergence of the marginal distribution of the model toward the stationary distribution is slow. It is therefore necessary to use this single sequence approximation with extreme caution when the stationary assumption of the model is clearly in contradiction with the observed data.

Thanks to Algorithm 1, it is possible for the first time (up to our knowledge) to study the distribution of patterns in a data set of upstream regions using an heterogeneous model. Despite the fact that there are some noticeable differences between this heterogeneous model and its homogeneous alternative, in practice we observe very little difference between the resulting P-values for most of the tested patterns. Some patterns are nevertheless more sensitive than others to the heterogeneity of the data, and their P-values may by altered by a factor 10 or more.

It should also be noted that heterogeneous Markov chains may be used to describe the behavior of homogeneous Markov chains under particular constraints. For example, this is exactly the distribution we get when considering the distribution of the hidden sequence of a HMM conditionally to the observed data (e.g., detection of CpG islands [20]). We get similar distribution when we take into account the special characters (`N` means "any nucleotides" in DNA sequences; `X` means "any aminoacid" in proteins) in biological sequences [21].

There are several interesting directions for further developments of this work. The first one could be to slightly change the statistic of interest for patterns problem by considering the $M = M_1 + ... + M_r$ number of matching sequences instead of the number of occurrences. Such a choice might be motivated by the nature of the selection pressure on a particular pattern: at least k occurrences of the pattern in a sequence insure a given biochemical activity (e.g., structured motifs in regulation [57]). In such a case, the pattern would match sequence j ($M_j = 1$) if it occurs at least k times in the sequence, and would else mismatch the sequence ($M_j = 0$). From a technical point of view, this is only a minor extension of the present work, where one only needs to adapt the existing method to get the moment generating function of each M_j. However, the practical interest of such alternative statistic for pattern problem is yet to be studied.

A open problem remains open: how to deal with high complexity patterns (high L) in long homogeneous sequences (high ℓ)? The partial recursion we introduce here might be a solution, but it is necessary to study in further details its numerical stability issues. The only alternative seems to be the sparse LU bivariate polynomial approach suggested above to compute the bivariate moment generating function $G(y, z)$. However, an exhaustive study of the relation between pattern complexity and the denominator degree of $G(y, z)$ remains to be done in order to assess the practical interest of this approach.

Finally, let us point out that all the methods and algorithms presented in this paper are not yet available in an efficient implementation. One important task yet to be completed is to add these innovative techniques into the Statistics for Patterns package (SPatt) the purpose of which is to gather and make available the best pattern methods. SPatt is a C++ General Public License (GPL) program package which is freely available at the following url: http://stat.genopole.cnrs.fr/spatt

Acknowledgements
We are grateful to the anonymous referee for their extensive and constructive remarks and comments.

Author details
[1]LSG, Laboratoire Statistique et Génome, CNRS UMR-8071, INRA UMR-1152, University of Evry, Evry, France. [2]CNRS, Paris, France. [3]MAP5, Department of Applied Mathematics, CNRS UMR-8145, University Paris Descartes, Paris, France. [4]EBGM, Equipe de Bioinformatique Génomique et Moleculaire, INSERM UMRS-726, University Paris Diderot, Paris, France. [5]MTi, Molecules Thérapeutique in silico, INSERM UMRS-973, University Paris Diderot, Paris, France. [6]MIG, Mathématique Informatique et Genome, INRA UR-1077, Jouy-en-Josas, France. [7]IBCP, Institut de Biologie et Chimie des Protéines, IFR 128, CNRS UMR 5086, University of Lyon 1, Lyon, France.

Authors' contributions
GN developed the statistical results and algorithms and carried out their implementation and application with heterogeneous models. LR was in charge of the application to structural motifs in protein loops. JM was in charge of the PROSITE application and of the study of DNA upstream regions with homogeneous models. The redaction of the paper have been done by ACC and GN. All authors read an approved the final manuscript.

Competing interests
The authors declare that they have no competing interests.

Received: 22 September 2009
Accepted: 26 January 2010 Published: 26 January 2010

References

1. Liolios K, Mavromatis K, Tavernarakis N, Kyrpides NC: **The Genomes On Line Database (GOLD) in 2007: status of genomic and metagenomic projects and their associated metadata.** *Nucleic Acids Res* 2008, **36**:D475-479.
2. Benson DA, Karsch-Mizrachi I, Lipman DJ, Ostell J, Sayers EW: **GenBank.** *Nucleic Acids Res* 2009, **37**:26-31.
3. Bairoch A, Bougueleret L, Altairac S, Amendolia V, Auchincloss A, Argoud-Puy G, Axelsen K, Baratin D, Blatter MC, Boeckmann Bea: **The Universal Protein Resource (UniProt) 2009.** *Nucleic Acids Res* 2009, **37**:D169-174.
4. Leung MY, Marsh GM, Speed TP: **Over and underrepresentation of short DNA words in Herpesvirus genomes.** *J Comp Biol* 1996, **3**:345-360.
5. Rocha E, Viari A, Danchin A: **Oligonucleotide bias in Bacillus subtilis: general trends and taxonomic comparisons.** *Nucl Acids Res* 1998, **26**:2971-2980.
6. Karlin S, Burge C, Campbell A: **Statistical analyses of counts and distributions of restriction sites in DNA sequences.** *Nucl Acids Res* 1992, **20(6)**:1363-1370.
7. Sourice S, Biaudet V, El Karoui M, Ehrlich S, Gruss A: **Identification of the Chi site of Haemophilus influenzae as several sequences related to Escherichia coli Chi site.** *Mol Microbiol* 1998, **27**:1021-1029.
8. Van Helden J, Olmo M, Perez-Ortin JE: **Statistical analysis of yeast genomic downstream sequences revels putative polyadenylation signals.** *Nucl Acids Res* 2000, **28(4)**:1000-1010.
9. Hulo N, Bairoch A, Bulliard V, Cerutti L, Cuche BA, de Castro E, Lachaize C, Langendijk-Genevaux PS, Sigrist CJ: **The 20 years of PROSITE.** *Nucleic Acids Res* 2008, **36**:D245-249.
10. Stormo GD: **DNA binding sites: representation and discovery.** *Bioinformatics* 2000, **16**:16-23.
11. Claverie JM, Audic S: **The statistical significance of nucleotide position-weight matrix matches.** *Comput Appl Biosci* 1996, **12**:431-439.
12. Frith MC, Spouge JL, Hansen U, Weng Z: **statistical significance of clusters of motifs represented by position specific scoring matrices in nucleotide sequences.** *Nuc Acids Res* 2002, **30(14)**:3214-3224.
13. Gautier C: **Compositional bias in DNA.** *Curr Opin Genet Dev* 2000, **10**:656-661.
14. Nicolas P, Bize L, Muri F, Hoebeke M, Rodolphe F, Ehrlich S, Prum B, Bessières P: **Mining Bacillus subtilis chromosome heterogeneities using hidden Markov models.** *Nucleic Acids Res* 2002, **30**:1418-1426.
15. Do J, Choi D: **Computational approaches to gene prediction.** *J Microbiol* 2006, **44**:137-144.
16. Becq J, Gutierrez M, Rosas-Magallanes V, Rauzier J, Gicquel B, Neyrolles O, Deschavanne P: **Contribution of horizontally acquired genomic islands to the evolution of the tubercle bacilli.** *Mol Biol Evol* 2007, **24**:1861-1871.
17. Martin J, Gibrat J, Rodolphe F: **Analysis of an optimal hidden Markov model for secondary structure prediction.** *BMC Struct Biol* 2006, **6**:25.
18. Churchill G: **Stochastic models for heterogeneous DNA sequences.** *Bull Math Biol* 1989, **268**:8-14.
19. Fickett JW, Torney DC, Wolf DR: **Base compositional Structure of Genomes.** *Genomics* 1992, **13**:1056-1064.
20. Aston JAD, Martin DEK: **Distributions associated with general runs and patterns in hidden Markov models.** *Ann Appl Stat* 2007, **1**:585-61.
21. Nuel G: **Couting patterns in degenerated sequences.** *PRIB 2009, of Lec. Notes in Bioinfo* 2009, **5780**:222-232.
22. Reignier M: **A unified approach to word occurrences probabilities.** *Discrete Applied Mathematics* 2000, **104**:259-280.
23. Reinert G, Schbath S: **Probabilistic and Statistical Properties of Words: An Overview.** *J of Comp Biol* 2000, **7(1-2)**:1-46.
24. Lothaire M, Ed: *Applied Combinatorics on Words* Cambridge University Press, Cambridge 2005.
25. Nuel G: **Numerical solutions for Patterns Statistics on Markov chains.** *Stat App in Genet and Mol Biol* 2006, **5**:26.
26. Fu JC: **Distribution theory of runs and patterns associated with a sequence of multi-state trials.** *Statistica Sinica* 1996, **6(4)**:957-974.
27. Stefanov V, Pakes AG: **Explicit distributional results in pattern formation.** *Ann Appl Probab* 1997, **7**:666-678.
28. Antzoulakos DL: **Waiting times for patterns in a sequence of multistate trials.** *J Appl Prob* 2001, **38**:508-518.
29. Chang YM: **Distribution of waiting time until the rth occurrence of a compound pattern.** *Statistics and Probability Letters* 2005, **75**:29-38.
30. Boeva V, Clément J, Régnier M, Vandenbogaert M: **Assessing the significance of Sets of Words.** *Combinatorial Pattern Matching 05, Lecture Notes in Computer Science, Springer-Verlag* 2005, **3537.**
31. Nuel G: **Effective p-value computations using Finite Markov Chain Imbedding (FMCI): application to local score and to pattern statistics.** *Algorithms for Molecular Biology* 2006, **1**:5.
32. Stefanov VT, Szpankowski W: **Waiting Time Distributions for Pattern Occurrence in a Constrained Sequence.** *Discrete Mathematics and Theoretical Computer Science* 2007, **9**:305-320.
33. Boeva V, Clement J, Regnier M, Roytberg M, Makeev V: **Exact p-value calculation for heterotypic clusters of regulatory motifs and its application in computational annotation of cis-regulatory modules.** *Algorithms for Molecular Biology* 2007, **2**:13.
34. Pevzner P, Borodovski M, Mironov A: **Linguistic of nucleotide sequences: The significance of deviation from mean statistical characteristics and prediction of frequencies of occurrence of words.** *J Biomol Struct Dyn* 1989, **6**:1013-1026.
35. Cowan R: **Expected frequencies of DNA patterns using Whittle's formula.** *J Appl Prob* 1991, **28**:886-892.
36. Kleffe J, Borodovski M: **First and second moment of counts of words in random texts generated by Markov chains.** *Bioinformatics* 1997, **8(5)**:433-441.
37. Prum B, Rodolphe F, de Turckheim E: **Finding words with unexpected frequencies in DNA sequences.** *J R Statist Soc B* 1995, **11**:190-192.
38. Godbole AP: **Poissons approximations for runs and patterns of rare events.** *Adv Appl Prob* 1991, **23.**
39. Geske MX, Godbole AP, Schaffner AA, Skrolnick AM, Wallstrom GL: **Compound Poisson approximations for word patterns under Markovian hypotheses.** *J Appl Probab* 1995, **32**:877-892.
40. Reinert G, Schbath S: **Compound Poisson and Poisson process approximations for occurrences of multiple words in markov chains.** *J of Comp Biol* 1999, **5**:223-254.
41. Erhardsson T: **Compound Poisson approximation for counts of rare patterns in Markov chains and extreme sojourns in birth-death chains.** *Ann Appl Probab* 2000, **10(2)**:573-591.
42. Nuelg G: **Cumulative distribution function of a geometric Poisson distribution.** *J Stat Comp and Sim* 2008, **78(3)**:211-220.
43. Denise A, Régnier M, Vandenbogaert M: **Assessing the Statistical Significance of Overrepresented Oligonucleotides.** *Lecture Notes in Computer Science* 2001, **2149**:85-97.
44. Nuel G: **LD-SPatt: Large Deviations Statistics for Patterns on Markov Chains.** *J Comp Biol* 2004, **11(6)**:1023-1033.
45. Fu J, Johnson B: **Approximate Probabilities for Runs and Patterns in i.i.d. and Markov Dependent Multi-state Trials.** *Adv in Appl Prob* 2009, **41**:292-308.
46. Nicodème P, Salvy B, Flajolet P: **Motif statistics.** *Theoretical Com Sci* 2002, **287(2)**:593-617.

47. Crochemore M, Stefanov V: **Waiting time and complexity for matching patterns with automata.** *Info Proc Letters* 2003, **87(3)**:119-125.
48. Lladser ME: **Mininal Markov chain embeddings of pattern problems.** *Information Theory and Applications Workshop* 2007, 251-255.
49. Nuel G: **Pattern Markov chains: optimal Markov chain embedding through deterministic finite automata.** *J of Applied Prob* 2008, **45**:226-243.
50. Ribeca P, Raineri E: **Faster exact Markovian probability functions for motif occurrences: a DFA-only approach.** *Bioinformatics* 2008, **24(24)**:2839-2848.
51. Nuel G: **On the first *k* moments of the random count of a pattern in a multi-states sequence generated by a Markov source.**http://arxiv.org/pdf/0909.4071, ArXiv.
52. Fu JC, Koutras MV: **Distribution theory of runs: a Markov chain approach.** *J Amer Statist Assoc* 1994, **89**:1050-1058.
53. Camproux AC, Gautier R, Tufféry T: **A hidden Markov model derivated structural alphabet for proteins.** *J Mol Biol* 2004, **339**:561-605.
54. Regad L, Martin J, Camproux AC: **Identification of non Random Motifs in Loops Using a Structural Alphabet.** *Proceedings of IEEE Symposium on Computational Intelligence in Bioinformatics and Computational* 2006, 92-100.
55. Hopcroft JE, Motwani R, Ullman JD: *Introduction to Automata Theory, Languages, and Computation* Addison-Wesley 2006.
56. Thomas-Chollier M, Sand O, Turatsinze JV, Janky R, Defrance M, Vervisch E, Brohée S, van Helden J: **RSAT: regulatory sequence analysis tools.** *Nucleic Acids Res* 2008, **36**:W119-127.
57. Stefanov V, Robin S, Schbath S: **Waiting times for clumps of patterns and for structured motifs in random sequences.** *Discrete Applied Mathematics* 2007, **155**:868-880.

doi:10.1186/1748-7188-5-15
Cite this article as: Nuel *et al.*: Exact distribution of a pattern in a set of random sequences generated by a Markov source: applications to biological data. *Algorithms for Molecular Biology* 2010 **5**:15.