Mining for unexpected sequential patterns given a Markov model

Cécile Low-Kam, André Mas, Maguelonne Teisseire

Mining for unexpected sequential patterns given a Markov model

Cécile Low-Kam André Mas
I3M, Univ. Montpellier 2 - CNRS
clowkam, mas@math.univ-montp2.fr

Maguelonne Teisseire
LIRMM, Univ. Montpellier 2 - CNRS
teisseire@lirmm.fr

Abstract

Large databases are now available and many works offer to mine for the information within. Nevertheless, extracting interesting sites from biological sequential databases remains a challenging task, as they are known to occur with errors. Sequential patterns, by their flexible structure, enable to overcome this problem. However, those patterns are usually numerous while many of them are not relevant. Therefore, in this paper, we propose a new approach to mine for statistically significant sequential patterns. It extracts fewer patterns than more traditional approaches. Experiments show that such sequential patterns are pertinent for biological databases.

1 Introduction

Recently, many works have focused on identifiying patterns that may have a biological function, in sequences of DNA or proteins. Some approaches aim at finding such motifs among DNA sequences, but focus on a restricted kind of patterns called *"structured motifs"* [Robin *et al.*, 2002], [Zhang *et al.*, 2007]. Those are made of two "boxes" of nucleotides at a bounded distance from each other. Structured motifs having an unexpected frequency given a chosen model are mined. Those patterns cover many regions of interest in DNA sequences. For example, in promoter sequences, the TATA box is a well-known binding site. However, structured motifs are too specific to be general and do not extend to other areas of biological function, such as motifs in protein primary structure. To this end, alternative structures are needed.

Sequential patterns, first introduced in [Agrawal and Srikant, 1995], describe frequent rules of behaviour on ordered sequences of events. For example, in a database of DNA sequences, one such rule would be *"In 80% of sequences the nucleotide T occurs, followed some time later by A, then T, and then A."*. Therefore, there are gaps of variable length between two elements of the pattern. But those patterns are usually numerous, making their interpretation difficult. Consequently, many works reduce the redundancy of the discovered patterns, for example through condensed representations [Mannila and Toivonen, 1996]. However, our aim here is to extract patterns that are not only frequent, but also may have a role in genetics.

s	*A*	*T*	*G*	G	A
t	*A*	C	*T*	*G*	A
u	C	T	T	G	A

Table 1: An example of DNA database.

Therefore, in the present work, we mine for frequent sequential patterns that occur unexpectedly given a model. Sequential patterns are more general and flexible than structured motifs, therefore we are able to highlight other regions of interest in biological sequences. As databases of DNA or proteins are made of long sequences on a small alphabet [Wang *et al.*, 2004], many sequential patterns would be outlined by a basic extraction. Our approach reduces the set of patterns so as to get fewer but of biological interest.

The rest of the paper is organised as follows: problem definition is stated in Section 2. Our approach and the associated algorithm is presented in Section 3. Experiments are led in Section 4. Related work is recalled in Section 5. Conclusions and further work are found in Section 6.

2 Problem definition

In this section, we present the problem of mining for unexpected sequential patterns. Let us consider a database of ordered transactions over a finite alphabet Σ of items. To each transaction correspond a sequence identifiant, an order identifiant and an item. A sequence of items is called a *sequence of data*. For example, Table 1 contains 3 sequences of data on the alphabet $\{A, T, G, C\}$. In this context, a subsequence $x = \langle x_1 ... x_k \rangle$ is said to be *included* in a sequence of data $s = s_1 ... s_\ell$ if there are integers $1 \leq i_1 < ... < i_k \leq \ell$ such that $s_{i_1} = x_1, ..., s_{i_k} = x_k$. That is, items are not necessarily consecutive. In the previous example, the subsequence $\langle ATG \rangle$ is included in the sequence of data s, but also in t (even if there is a gap between A and T). However, it is not included in u. A sequence s *supports* a subsequence x if x is included in s. The number of sequences supporting a subsequence x is called the *support* of x and is denoted by $Supp(x)$. Let $minSupp$ be the minimal support chosen by the user. Then all subsequences x such that $Supp(x) \geq minSupp$ are called *frequent*. For example, if

we set $minSupp = 2$, the subsequence $\langle ATG \rangle$ is frequent.

Mining for sequential patterns aims at discovering all frequent subsequences, given a threshold $minSupp$. Let us note that we have restricted the notion of sequential pattern to the specific context of sequences of items. Usually, sequential patterns are more general, as in [Agrawal and Srikant, 1995], they were introduced on databases of ordered sets of items. Over the last few years, many algortihms have been developed to extract sequential patterns from databases of transactions. As the number of potential sequential patterns may be huge, such algorithms use the anti-monotonic property to be able to extract them. However, the object of this work is to mine patterns which besides being frequent, also might have a biological function. To do so, we fit a model which captures the global characteristics of the data, and look for sequential patterns which frequencies are not well explained by this model. More precisely, we outline sequential patterns which observed frequencies under the model are significantly greater than expected. This is done by adding a filtering step to existing algorithms. We use dynamic programming to compute expected supports.

3 Unexpected Sequential Patterns

In this section, we describe how frequencies of sequential patterns according to a model are computed and compared to observed frequencies, so as to extract patterns of interest.

3.1 Sequential Patterns under a Markov Assumption

A Markov chain or process $S_1, S_2, ...$ is characterised by the following property: for $i \geq 2$,

$$Pr\left(S_i = s_i | S_1^{i-1} = s_1^{i-1}\right) = Pr(S_i = s_i | S_{i-1} = s_{i-1}), \quad (1)$$

where $S_1^{i-1} = (S_1, ..., S_{i-1})$ and $s_1^{i-1} = (s_1, ..., s_{i-1})$. It is a Markov property of order 1 but it can easily be extended to greater orders. The interest of the Markov model is twofold: first, it is adapted for many sequences such as DNA or proteins, since it takes a temporal dependency into account, and second, it involves a small set of parameters thanks to the reduced dependence.

We consider a database DB of n independant realisations of a Markov process $\{S_t, t > 0\}$ on a finite state space $\Sigma = \left\{\sigma_1, ..., \sigma_{|\Sigma|}\right\}$. One such realisation is a sequence $s = s_1 ... s_\ell$. We also suppose that the process is time-homogeneous, that is, for all $1 \leq i$,

$$Pr(S_i = s_i | S_{i-1} = s_{i-1}) = Pr(S_2 = s_2 | S_1 = s_1)\,. \quad (2)$$

Moreover, we assume that the process has reached a stationary distribution, *i.e.* for $i_1 < ... < i_j$ and $k > 0$:

$$Pr\left(S_{i_1}^{i_j} = s_{i_1}^{i_j}\right) = Pr\left(S_{i_1+k}^{i_j+k} = s_{i_1}^{i_j}\right). \quad (3)$$

Consequently, $\{S_t, t > 0\}$ is characterised by a stationary distribution μ of length $|\Sigma|$ which k^{th} element denoted by μ_k is equal to $Pr(S_1 = \sigma_k)$, and a transition matrix P of size $|\Sigma|$ which $(j,k)^{th}$ element denoted by P_{jk} is equal to $Pr(S_2 = \sigma_k | S_1 = \sigma_j)$.

S_1	S_2	S_3	S_4
x_1	x_2	.	.
x_1	$\{x_2\}^c$	x_2	.
x_1	$\{x_2\}^c$	$\{x_2\}^c$	x_2
$\{x_1\}^c$	x_1	x_2	.
$\{x_1\}^c$	$\{x_1\}^c$	x_1	x_2

Table 2: Possible configurations for x in S.

We now define what a sequential pattern is, according to this model. Let us consider a subsequence $x = \langle x_1 ... x_k \rangle$. The probability for a sequence S of length ℓ to support x is denoted by $p_\ell(x)$ as it depends on the length of the sequence. To calculate $p_\ell(x)$, we enumerate all sequences in which x may appear. To do so, we enumerate all possible occurences of x according to the first occurence of each of its letter. For each $\sigma \in \Sigma$, we denote by $\{\sigma\}^c$ the set $\{\rho \in \Sigma, \rho \neq \sigma\}$. The character "." is the set of all possible items $\{\sigma, \sigma \in \Sigma\}$.

To fix the ideas, let us consider a random sequence $S = S_1 ... S_4$ of length 4, and a pattern $x = \langle x_1 x_2 \rangle$ of length 2. Let us suppose that S supports x. Then S may be written as one and only one of the lines of Table 2. Let us suppose that $S_1 = x_1$. The first possible configuration is that $S_2 = x_2$ (first line of Table 2). Let us note that x_1 and x_2 may occur further on the sequence. The next possible configuration is $S_2 \neq x_2$ but $S_3 = x_2$ (second line). This configuration do not overlap on the previous one. The last possibility is $S_2 \neq x_2$, $S_3 \neq x_2$ but $S_4 = x_2$ (third line). Let us now suppose that $S_1 \neq x_2$ and $S_2 = x_1$. We carry on with the enumeration until the last possible configuration. Thus all possibilities are counted, without redundancy. This manner of counting is easily extended to the case of a pattern of length k and a sequence of length ℓ, where $\ell \geq k$. In order to write it properly we introduce the two following probabilities.

- For $1 \leq k \leq |\Sigma|$, $q_{\sigma_k}(i)$ is the probability that the first occurence of σ_k is at the i^{th} position of the sequence:

$$\begin{cases} q_{\sigma_k}(1) = Pr\left(S_1 = \sigma_k\right), \\ q_{\sigma_k}(i) = Pr\left(S_i = \sigma_k, S_m \neq \sigma_k, 1 \leq m < i\right), \\ \text{for } 1 < i \leq \ell. \end{cases} \quad (4)$$

  More precisely, and using the notations defined at the beginning of this section,

$$\begin{cases} q_{\sigma_k}(1) = \mu_k, \\ q_{\sigma_k}(2) = \sum\limits_{j \neq k} \mu_j P_{jk}, \\ q_{\sigma_k}(i) = \sum\limits_{j_1, ..., j_{i-1} \neq k} \mu_{j_1} \prod\limits_{m=1}^{i-2} P_{j_m, j_{m+1}} P_{j_{i-1} k}, \\ \text{for } 3 \leq i \leq \ell. \end{cases} \quad (5)$$

- For $1 \leq j, k \leq |\Sigma|$, $q_{\sigma_j \sigma_k}(i)$ is the probability that σ_k occurs at the i^{th} position after σ_j, and not in between.

By stationarity,

$$\begin{cases} q_{\sigma_j\sigma_k}(1) = Pr\left(S_2 = \sigma_k | S_1 = \sigma_j\right), \\ q_{\sigma_j\sigma_k}(i) = Pr(S_{i+1} = \sigma_k, S_m \neq \sigma_k, 1 < m \leq i | S_1 = \sigma_j), \\ \text{for } 1 < i \leq \ell - 1. \end{cases} \tag{6}$$

Therefore, using the notations for the stationary distribution and the transition matrix of the Markov chain,

$$\begin{cases} q_{\sigma_j\sigma_k}(1) = P_{jk}, \\ q_{\sigma_j\sigma_k}(2) = \sum\limits_{r \neq k} P_{jr} P_{rk}, \\ q_{\sigma_j\sigma_k}(i) = \sum\limits_{r_1,\ldots,r_{i-1} \neq k} P_{jr_1} \prod\limits_{m=1}^{i-2} P_{r_m r_{m+1}} P_{r_{i-1}k}, \\ \text{for } 3 \leq i \leq \ell - 1. \end{cases} \tag{7}$$

Let us now describe how to compute these quantities. For the sake of simplicty, we only present here how to obtain (5) as we proceed identically for the probabilities of (7). The transpose of any vector v is denoted by v^T. $\times$ denotes the cross product, and for $1 \leq j \leq |\Sigma|$, $P_{j.}$ is the j^th line of P.

Algorithm 1: First Occurence

Data: An index k, a stationary distribution μ, a transition probability P.
Result: $\{q_{\sigma_k}(i), 1 \leq i \leq \ell\}$.

```
Initialisation
Compute μ
Compute P
q_{σ_k}(1) ← μ_k
ν ← μ
π ← (P_{1.}, ..., P_{k-1.}, 0, P_{k+1.}, ..., P_{|Σ|.})^T
for 2 ≤ i ≤ ℓ do
    ν ← ν × π
    q_{σ_k}(i) ← ν_k
end
```

First, the stationary distribution μ and the transition matrix P are computed. This is done by scanning twice the database DB, resulting in a complexity of $O(2n\ell)$, if ℓ denotes the average length of sequences of DB. The matrix π of Algorithm 1 is similar to P except for the k^{th} line which is null. This secures that the state σ_k will not be reached until the i^{th} transaction. The complexity of Algorithm 1 is $O(\ell\,|\Sigma|^2)$. The transition probabilities $q_{\sigma_j\sigma_k}$ are computed in a similar way for $1 \leq j, k \leq |\Sigma|$, with a total complexity of $O(\ell\,|\Sigma|^3)$.

Counting all possible occurences, the probability for x to occur in S is thus:

$$p_\ell(x) = \sum_{i_1 \in I_1, i_2 \in I_2, \ldots, i_k \in I_k} q_{x_1}(i_1) q_{x_1x_2}(i_2) \ldots q_{x_{k-1}x_k}(i_k), \tag{8}$$

where $I_j = \{1, 2, \ldots, \ell + j - k - 1 - i_{j-1}\}$. Therefore, the probability of occurence $p_\ell(x)$ of each pattern x under the Markov model can be calculated explicitely. However, a naive computation would have poor time efficiency. Therefore, in the next subsection, we describe how to deduce the support of any pattern given the support of its greater prefix. This property allows to reduce the computation time like other dynamic programming approaches, such as the well-known Forward algorithm.

3.2 Recurrence

For $x = \langle x_1 \ldots x_k \rangle$, let $Q_x(i)$ be the probability that the first occurence of x ends at the i^{th} position of the sequence. Let Q_x be the vector of length $\ell - |x| + 1$ such that $Q_x = (Q_x(i))_{|x| \leq i \leq \ell}$. Then, according to the enumeration we have presented, the sum of the terms of Q_x is $p_\ell(x)$. Indeed, the probability for one pattern to appear at least once in a sequence is the sum of the probabilities of its first occurence at any position.

- $k = 1$: then x is reduced to a single letter and $Q_x = (q_x(i))_{1 \leq i \leq \ell}$ is obtained as in (5).
- $1 < k \leq \ell$: then if x_- denotes $\langle x_1 \ldots x_{k-1} \rangle$, the largest prefix of x, then for $k \leq i \leq \ell$,

$$Q_x(i) = \sum_{j=|x|-1}^{i-1} Q_{x_1 \ldots x_{k-1}}(j)\; q_{x_{k-1}x_k}(i-j). \tag{9}$$

Thus, (9) allows to use previous calculations. Next, we describe more precisely how (9) is done. Let M_{x_-} be the triangular matrix defined as follows:

$$M_{x_-} = \begin{pmatrix} Q_{x_-}(|x|) & Q_{x_-}(|x|+1) & \cdots & Q_{x_-}(\ell-1) \\ 0 & Q_{x_-}(|x|) & \cdots & Q_{x_-}(\ell-2) \\ 0 & 0 & \cdots & Q_{x_-}(\ell-3) \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & Q_{x_-}(|x|) \end{pmatrix}. \tag{10}$$

Then, Q_x is obtained as

$$M_{x_-} \times \begin{pmatrix} q_{x_{k-1}x_k}(1) \\ q_{x_{k-1}x_k}(2) \\ \vdots \\ q_{x_{k-1}x_k}(\ell - |x| + 1) \end{pmatrix}. \tag{11}$$

Once the support of its greater suffix has been computed, the complexity of the calculation of the expected support of a pattern is reduced, as only its two last items' positions are enumerated, whereas the naive implementation would have considered all. Thus, we compute expected supports according to a Markov Model. Let us remark that those supports could be approximated by simulating sequences according to the parameters of the model. However, a large number of such simulations would be required to get accurate results. In order to extract patterns of interest, we now compare expected frequencies to observed frequencies.

3.3 Statistically Significant Patterns

For the sake of simplicty, we now consider that all sequences of DB have the same length ℓ. Our approach may be straightforwardly extended to databases of sequences of different

lengths. The support of a pattern x in a database DB of n independent sequences of data of length ℓ is a random variable of binomial distribution. Indeed, for any pattern x, and $1 \leq i \leq n$, let Z_x^i be the Bernoulli random variable such that:

$$\begin{cases} Z_x^i = 1 \text{ if the } i^{th} \text{ sequence supports x,} \\ Z_x^i = 0 \text{ if not,} \end{cases} \quad (12)$$

that is, $Z_x^i = 1$ with the probability $p_\ell(x)$. Then $Supp(x)$ is the sum of those independant identically distributed (*i.i.d.*) random variables of Bernoulli, and

$$Supp(x) \sim \mathcal{B}in(p_\ell(x), n). \quad (13)$$

Therefore, $n \times p_\ell(x) = \mathbb{E}(Supp(x))$ is the expected support of x. It is compared to the observed support $Supp_{obs}(x)$ by the means of the p-value:

$$Pr\{\mathcal{B}in(p_\ell(x), n) \geq Supp_{obs}(x)\}. \quad (14)$$

The pattern x is over-represented if this p-value is less than a given threshold.

Let us now describe the algorithm *Un-SP* which extracts unexpected sequential patterns.

Given probabilites of first occurence, a minimum support $minSupp$, and a threshold ε for the p-values, *Un-SP* does a depth-first search of the prefix tree of sequential patterns. This is done through the function For each pattern x, if its observed support is greater than $minSupp$, its support according to the model is computed using the support of its greater prefix. Therefore, we need to store the vectors Q_y for each suffix y of x. However, those vectors are discarded as we progress further on the tree. If $p_x < \varepsilon$, x is an unexpected sequential pattern. *Un-SP* is described in 2. At the end of the algorithm, we call for two functions $new.pref$ and $new.letter$ which allow to go to the next branch. In summary, *Un-SP* is a classical depth-first algorithm for sequential pattern extraction but outlines only unexpected ones thanks to a filtering step.

Algorithm 2: Un-SP

Data: A database DB, Probabilities of first occurence $q_{\sigma_k}(i)$,$q_{\sigma_j\sigma_k}(i)$ for $1 \leq i \leq \ell$, $1 \leq j,k \leq |\Sigma|$, a minimal support $minSupp$ and a threshold ε
Result: A set of unexpected sequential patterns $\mathcal{X}$

```
Initialisation
pref ← " "
i ← 1
X ← φ.
while i ≤ |Σ| do
    x ← pref · Σ[i].
    if Supp_obs(x) ≥ minSupp & length(x) ≤ ℓ then
        i ← 1
        Compute its expected support n × p_ℓ(x)
        Compute the associated p-value
        if p-value ≤ ε then
            X ← X ∪ {x}
        end
    end
    else if i < |Σ| then
        i ← i + 1
    end
    else
        pref ← new.pref(pref)
        i ← new.letter(pref)
    end
end
return X.
```

4 Experiments

In this section, we describe our experiments on real biological datasets. The algorithm *Un-SP* has been implemented in R [Team, 2006]. The R package "Bio3D" [Grant *et al.*, 2006] has been used to read protein sequences in the FASTA format. The object of this section is to re-establish well-known results to demonstrate the pertinence of our approach.

4.1 DNA sequences

In Section 1, we have seen that structured motifs are regions of interest in prokaryotic promoters. However, highlighting similar regions in eukaryotic promoters is less straightforward, as they present more diverse structures. Therefore, we consider a database from the Eukaryotic Promoter Database [Praz *et al.*, 2002] on the alphabet $\{A, T, G, C\}$. It contains 186 sequences of plant promoters chromosomal genes of length 52 from position -50 to $+1$ relative to transcription start site. Unexpected sequential patterns for $minSupp = 90\%$ and $\varepsilon = 1 \times 10^{-2}$ are presented in Table 3. We recognise the well-known $TATA$ box (sequence $TATAAA$), which often lies close to the transcription start site in eukaryotic promoters. Moreover, the set of discovered patterns is strongly reduced thanks to the constraint on the p-value. Figure 1 shows, using a logarithmic scale, the number of sequential patterns ($\varepsilon = 1$), the number of unexpected sequential patterns for $\varepsilon = 1 \times 10^{-1}, 1 \times 10^{-2}, 1 \times 10^{-3}$. For example, if $minSupp = 80\%$, there are 367696 sequential patterns, but only 2082 of p-value less than 1×10^{-1}, 321 of p-value less than 1×10^{-2}, and 57 of p-value less than 1×10^{-3}. Therefore, we have restricted the set of extracted sequential patterns to those which present a biological interest. We have found similar results than existing approaches focusing on structured motifs, as they are a particular kind of sequential patterns.

Pattern	p-value
$\langle CATAATAAAATCA \rangle$	0.007
$\langle CATATATAAAACA \rangle$	0.005
$\langle CATTATAAAAACA \rangle$	0.002
$\langle CTATATAAAATCA \rangle$	0.008
$\langle CTTATAAAAATCA \rangle$	0.004
$\langle CTTATAAAAACA \rangle$	0.007
$\langle CCTTATAAATTCA \rangle$	0.001

Table 3: $minSupp = 90\%$ and $\varepsilon = 1 \times 10^{-2}$

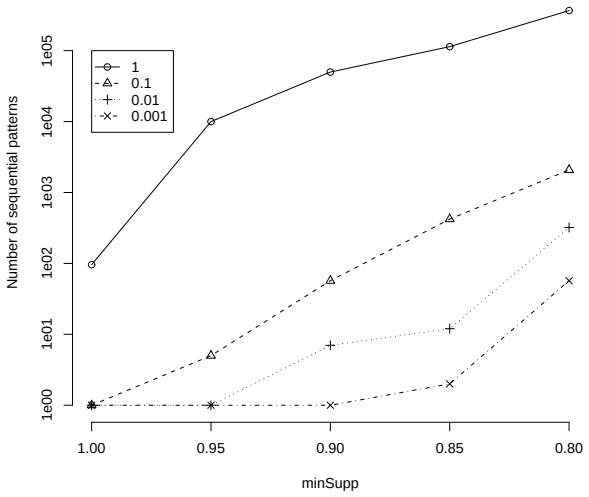

Figure 1: Number of sequential patterns extracted.

Pattern	p-value
$\langle CCFH \rangle$	1.74×10^{-84}
$\langle CCFHH \rangle$	1.29×10^{-112}
$\langle CCH \rangle$	2.03×10^{-53}
$\langle CCHH \rangle$	1.18×10^{-70}
$\langle CCKH \rangle$	1.40×10^{-58}
$\langle CCKHH \rangle$	2.44×10^{-71}
$\langle CCLH \rangle$	3.08×10^{-75}
$\langle CCLHH \rangle$	1.10×10^{-90}
$\langle CCRH \rangle$	5.44×10^{-61}
$\langle CCSH \rangle$	6.37×10^{-55}
$\langle CFHH \rangle$	3.65×10^{-67}

Table 4: $minSupp = 65\%$ and $\varepsilon = 1 \times 10^{-50}$

4.2 Protein sequences

Let us now consider protein sequences on the alphabet of 20 amino acids. *Domains* are defined as parts of protein sequences which, roughly, fold in a compact structure and are associated to a biological function. Many proteins include different domains. In every domain are found recognisable motifs. For example, there are different types of *Zinc Finger Motifs*, with specific compositions. For the C2H2 type, the consensus sequence is C-x$(2-4)$-C-x(3)-[LIVMFYWC]-x(8)-H-x$(3,5)$-H, which means that Coccurs, then 2 to 4 bases later C again, then 3 bases later any of the set $\{L, I, V, M, F, Y, W, C\}$, etc... We consider a dataset of 56 sequences of length 25 from the domain zf-C2H2 from the Pfam database [Bateman *et al.*, 2000], and check wether the consensus sequence is outlined by our approach. Results are presented in Table 4.

We recognise the motif CCHH at line 4 of Table 4, but also many parts of the consensus sequence. Therefore, we identify a well-known motif as the flexible structure of sequential patterns allows to consider sequences of variable length and gaps.

5 Related Work

In data mining, mining for unexpected sequential patterns has recently raisen interest. In [Li *et al.*, 2008], the notion of belief is introduced as a user-defined rule. Unexpected sequential patterns which contradict those beliefs, are then defined as unexpected. Experiments are led on a database of weblogs. However, this approach's object is to find sequential patterns which clash with beliefs defined by users, while we are interested in sequential patterns which are over-represented given a global model.

A similar problem to ours is studied in [Robin *et al.*, 2002] and [Zhang *et al.*, 2007]. Indeed, they aim at finding specific patterns called *structured motifs* of unexpected frequencies in a set of sequences. Structured motifs are formed of two boxes of consecutive nucleotides separated by a gap. For example, a structured motif in a DNA sequence would be $ATTG..TAGC$, where the two boxes $ATTG$ and $TAGC$ occur (possibly with errors) with two "do-not-care" symbols in-between. In [Robin *et al.*, 2002], the probability of occurence for one such motif at a given position is approximated by considering the past up to a fixed order. In [Zhang *et al.*, 2007], under a Markov assumption, the probability of occurence is computed by the inclusion-exclusion principle, but only for structured motifs with a fixed length gap. The probabilities of occurence are then compared to observed frequencies as the support follows a binomial distribution. However, these approaches are restricted to structured motifs, and cannot be applied to other patterns of biological function.

Other works study the problem of finding subsequences of consecutive letters of unexpected frequency given a model. In [Prum *et al.*, 1995], expected frequencies are computed given the exhaustive statistic of the model, and in [Flajolet *et al.*, 2006], by using generating functions. Let us notice that this last work also considers words formed of letters with variable-length gaps between them, called *"hidden patterns"* (similar to sequential patterns). In [Nuel, 2008], it is done by means of a convenient sequence associated to each word. Normal or Poisson approximations or Large Deviations allow to extract words for which expected frequencies differ significantly from observed ones. In [Jaroszewicz, 2008], subsequences which observed and predicted frequencies are larger than a given threshold, are used to update a hidden Markov model through dynamic programming. No p-value is introduced. However, those works differ from ours not only by the method used and the type of patterns considered, but also by the notion of frequency: for a given subsequence, all occurences in the database are counted. For example, in Table 1, the frequency of A would be $1/3$, as 5 out of 15 letters are A, whereas its support is 1, as it appears in 3 sequences out of 3. The same notion is used in [Califano, 2000] for identifying patterns with fixed gaps between their items, which frequency is greater than a given threshold.

In summary, existing works have identified various patterns of interest, in various databases. However, we present here an

approach to mine for unexpected sequential patterns under a probabilistic model assumption, which differs from those by the type of pattern, frequency, method or model considered.

6 Conclusion

In this paper, we have proposed a method to extract unexpected sequential patterns given a Markov Model. Expected supports have been computed thanks to an adequate enumeration. They have then been compared to observed supports by means of p-values. Computation time has been reduced thanks to dynamic programming. We have shown the practical interest of our approach by outlining patterns of interest in biological sequences. Future work will consist in extending our approach to include constraints on the maximal distance between occurences of items belonging to a same sequential pattern, such as in [Masseglia *et al.*, 2008].

References

[Agrawal and Srikant, 1995] Rakesh Agrawal and Ramakrishnan Srikant. Mining sequential patterns. In Philip S. Yu and Arbee S. P. Chen, editors, *Eleventh International Conference on Data Engineering*, pages 3–14, Taipei, Taiwan, 1995. IEEE Computer Society Press.

[Bateman *et al.*, 2000] A. Bateman, E. Birney, R. Durbin, S. R. Eddy, K. L. Howe, and E. L. Sonnhammer. The pfam protein families database. *Nucleic Acids Res.*, 28:263–266, 2000.

[Califano, 2000] Andrea Califano. Splash: structural pattern localization analysis by sequential histograms. *Bioinformatics*, 16(4):341–357, 2000.

[Flajolet *et al.*, 2006] Philippe Flajolet, Wojciech Szpankowski, and Brigitte Vallée. Hidden word statistics. *J. ACM*, 53(1):147–183, 2006.

[Grant *et al.*, 2006] B.J. Grant, A.P.C. Rodrigues, K.M. ElSawy, J.A. McCammon, and L.S.D. Caves. Bio3d: An r package for the comparative analysis of protein structures. *Bioinformatics*, 22:2695–2696, 2006.

[Jaroszewicz, 2008] S. Jaroszewicz. Interactive HMM construction based on interesting sequences. In *Proc. of Local Patterns to Global Models (LeGo'08) Workshop at the 12th European Conference on Principles and Practice of Knowledge Discovery in Databases (PKDD'08)*, pages 82–91, Antwerp, Belgium, 2008.

[Li *et al.*, 2008] Dong (Haoyuan) Li, Anne Laurent, and Pascal Poncelet. Mining unexpected web usage behaviors. In Petra Perner, editor, *ICDM*, volume 5077 of *Lecture Notes in Computer Science*, pages 283–297. Springer, 2008.

[Mannila and Toivonen, 1996] Heikki Mannila and Hannu Toivonen. Multiple uses of frequent sets and condensed representations. In *In Proc. KDD Int. Conf. Knowledge Discovery in Databases*, pages 189–194. AAAI Press, 1996.

[Masseglia *et al.*, 2008] Florent Masseglia, Pascal Poncelet, and Maguelonne Teisseire. Efficient mining of sequential patterns with time constraints: Reducing the combinations. *Expert Systems With Application*, 40:29, 2008.

[Nuel, 2008] Grgory Nuel. Pattern markov chains: optimal markov chain embedding through deterministic finite automata. *Journal of Applied Probability*, 1:226–243, 2008.

[Praz *et al.*, 2002] Vivivane Praz, Rouaida Perier, Claude Bonnard, and Philipp Bucher. The eukaryotic promoter database, epd: new entry types and links to gene expression data,. *Nuclear Acids Research*, 30:322–324, 2002.

[Prum *et al.*, 1995] B. Prum, F. Rodolphe, and . de Turckheim. Finding words with unexpected frequencies in DNA sequences. *J. R. Statist. Soc. B*, 57:205–220, 1995.

[Robin *et al.*, 2002] Stéphane Robin, Jean-Jacques Daudin, Hugues Richard, Marie-France Sagot, and Sophie Schbath. Occurrence probability of structured motifs in random sequences. *Journal of Computational Biology*, 9(6):761–774, 2002.

[Team, 2006] R Development Core Team. *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing, Vienna, Austria, 2006. ISBN 3-900051-07-0.

[Wang *et al.*, 2004] Ke Wang, Yabo Xu, and Jeffrey Xu Yu. Scalable sequential pattern mining for biological sequences. In *CIKM '04: Proceedings of the thirteenth ACM international conference on Information and knowledge management*, pages 178–187, New York, NY, USA, 2004. ACM.

[Zhang *et al.*, 2007] Jing Zhang, Bo Jiang, Ming Li, John Tromp, Xuegong Zhang, and Michael Q. Zhang. Computing exact P-values for DNA motifs. *Bioinformatics*, 23(5):531–537, 2007.