Mining for unexpected sequential patterns given a Markov model
Cécile Low-Kam, André Mas, Maguelonne Teisseire

To cite this version:
Cécile Low-Kam, André Mas, Maguelonne Teisseire. Mining for unexpected sequential patterns given a Markov model. 2008. <hal-00379780>
Mining for unexpected sequential patterns given a Markov model

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

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 $\Sigma$ of items. To each transaction correspond a sequence identifier, an order identifier 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 $x = (x_1...x_k)$ is said to be included in a sequence of data $s = s_1...s_t$ if there are integers $1 \leq i_1 \leq ... \leq i_k \leq t$ 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 $(ATG)$ 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 $\text{Supp}(x)$. Let $\text{minSupp}$ be the minimal support chosen by the user. Then all subsequences $x$ such that $\text{Supp}(x) \geq \text{minSupp}$ are called frequent. For example, if

<table>
<thead>
<tr>
<th>s</th>
<th>A</th>
<th>T</th>
<th>G</th>
<th>G</th>
<th>A</th>
</tr>
</thead>
<tbody>
<tr>
<td>t</td>
<td>A</td>
<td>C</td>
<td>T</td>
<td>G</td>
<td>A</td>
</tr>
<tr>
<td>u</td>
<td>C</td>
<td>T</td>
<td>G</td>
<td>A</td>
<td></td>
</tr>
</tbody>
</table>

Table 1: An example of DNA database.
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 algorithms 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 \( S_1, S_2, \ldots \) is characterised by the following property: for \( i \geq 2 \),

\[
Pr(S_i = s_i | S_{i-1} = s_{i-1}) = Pr(S_i = s_i | S_{i-2} = s_{i-2}),
\]

(1)

where \( S_{i-1} = (S_1, \ldots, S_{i-1}) \) and \( s_{i-1} = (s_1, \ldots, s_{i-1}) \). 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 \( \Sigma = \{ \sigma_1, \ldots, \sigma_{|\Sigma|} \} \). One such realisation is a sequence \( s = s_1 \ldots s_{\ell} \). We also suppose that the process is time-homogeneous, that is, for all \( 1 \leq i \),

\[
Pr(S_i = s_i | S_{i-1} = s_{i-1}) = Pr(S_2 = s_2 | S_1 = s_1).
\]

(2)

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

\[
Pr\left( S_{i_1}^j = s_{i_1}^j \right) = Pr\left( S_{i_1+k}^j = s_{i_1+k}^j \right).
\]

(3)

Consequently, \( \{ S_t, t > 0 \} \) is characterised by a stationary distribution \( \mu \) of length \( |\Sigma| \) which \( k^{th} \) element denoted by \( \mu_k \) 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 \( Pr(S_2 = \sigma_k | S_1 = \sigma_j) \).

We now define what a sequential pattern is, according to this model. Let us consider a subsequence \( x = (x_1, \ldots, x_k) \). The probability for a sequence \( S \) of length \( \ell \) to support \( x \) is denoted by \( p(x) \) as it depends on the length of the sequence. To calculate \( p(x) \), we enumerate all sequences in which \( x \) may appear. To do so, we enumerate all possible occurrences of \( x \) according to the first occurrence of each of its letter. For each \( \sigma \in \Sigma \), we denote by \( \{ \sigma \}^i \) 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 \( S = S_1 \ldots S_4 \) of length 4, and a pattern \( x = (x_1, x_2) \) 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 \( S_1 = x_2 \) (second line). This configuration do not overlap on the previous one. The last possibility is \( S_2 \neq x_2, S_1 \neq x_2 \) but \( S_4 = x_2 \) (third line). Let us now suppose that \( S_1 \neq x_2 \) and \( S_2 = x_2 \). 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 \( \ell \), where \( \ell > k \). 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 occurrence of \( \sigma_k \) is at the \( i^{th} \) position of the sequence:

\[
\begin{align*}
q_{\sigma_k}(1) & = Pr( S_1 = \sigma_k ), \\
q_{\sigma_k}(i) & = Pr( S_i = \sigma_k, S_{m} \neq \sigma_k, 1 \leq m < i ), \\
& \text{for } 1 < i \leq \ell.
\end{align*}
\]

(4)

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

\[
\begin{align*}
q_{\sigma_k}(1) & = \mu_k, \\
q_{\sigma_k}(2) & = \sum_{j \neq k} \mu_j P_{jk}, \\
q_{\sigma_k}(i) & = \sum_{j_1, \ldots, j_{i-1} \neq k} \mu_{j_1} \prod_{m=1}^{i-2} P_{j_m,j_{m+1}} P_{j_{i-1}k}, \\
& \text{for } 3 \leq i \leq \ell.
\end{align*}
\]

(5)

- For \( 1 \leq j, k \leq |\Sigma| \), \( q_{\sigma_j \sigma_k}(i) \) is the probability that \( \sigma_k \) occurs at the \( i^{th} \) position after \( \sigma_j \), and not in between.
By stationarity,
\[
\begin{align*}
q_{\sigma, \sigma_k}(1) &= P_r(S_2 = \sigma_k | S_1 = \sigma), \\
q_{\sigma, \sigma_k}(i) &= P_r(S_i = \sigma_k | S_{i-1} = \sigma, S_m \neq \sigma_k, 1 < m \leq i | S_1 = \sigma),
\end{align*}
\]
for 1 < i ≤ ℓ - 1.

Therefore, using the notations for the stationary distribution and the transition matrix of the Markov chain,
\[
\begin{align*}
q_{\sigma, \sigma_k}(1) &= P_{r_k}, \\
q_{\sigma, \sigma_k}(2) &= \sum_{r \neq k} P_{r_l} P_{r_k}, \\
q_{\sigma, \sigma_k}(i) &= \sum_{r_1, \ldots, r_{i-1} \neq k} P_{r_{j_1}} \prod_{m=1}^{i-2} P_{r_m r_{m+1}} P_{r_{i-1} k},
\end{align*}
\]
for 3 ≤ i ≤ ℓ - 1.

Let us now describe how to compute these quantities. For the sake of simplicity, 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 ≤ j ≤ |Σ|, \( P_j \) is the jth line of \( P \).

**Algorithm 1:** First Occurrence

**Data:** An index \( k \), a stationary distribution \( \mu \), a transition probability \( P \).

**Result:** \( \{q_{\sigma_k}(i), 1 \leq i \leq \ell\} \).

1. Initialization
2. Compute \( \mu \)
3. Compute \( P \)
4. \( q_{\sigma_k}(1) \leftarrow \mu_k \)
5. \( \nu \leftarrow (\mu_k) \)
6. \( \pi \leftarrow (P_1, \ldots, P_{k-1}, 0, P_{k+1}, \ldots, P_{|\Sigma|})^T \)
7. for \( 2 \leq i \leq \ell \) do
8. \( \nu \leftarrow \nu \times \pi \)
9. \( q_{\sigma_k}(i) \leftarrow \nu_k \)
10. end

First, the stationary distribution \( \mu \) 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 \( \ell \) denotes the average length of sequences of \( DB \). The matrix \( \pi \) of Algorithm 1 is similar to \( P \) except for the \( k^{th} \) line which is null. This ensures that the state \( \sigma_k \) will not be reached until the \( i^{th} \) transaction. The complexity of Algorithm 1 is \( O(\ell |\Sigma|^2) \). The transition probabilities \( q_{\sigma, \sigma_k} \) are computed in a similar way for 1 ≤ j, k ≤ |\Sigma|, with a total complexity of \( O(\ell |\Sigma|^3) \).

Counting all possible occurrences, the probability for \( x \) to occur in \( S \) is thus:
\[
p_x(x) = \sum_{i_1 \in I_1, i_2 \in I_2, \ldots, i_\ell \in I_\ell} q_{r_1}(i_1) q_{r_2}(i_2) \cdots q_{r_{\ell-1}}(i_{\ell-1}) q_{r_\ell}(i_\ell),
\]
where \( I_\ell = \{1, 2, \ldots, \ell + j - k - 1 - i_{j-1}\} \). Therefore, the probability of occurrence \( p_x(x) \) of each pattern \( x \) under the Markov model can be calculated explicitly. 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 \( x = (x_1 \ldots x_\ell) \), let \( Q_x(i) \) be the probability that the first occurrence of \( x \) ends at the \( i^{th} \) position of the sequence. Let \( Q_x \) be the vector of length \( \ell = |x| + 1 \) such that \( Q_x = (Q_x(i))_{0 \leq i \leq \ell} \). Then, according to the enumeration we have presented, the sum of the terms of \( Q_x \) is \( p_x(x) \). Indeed, the probability for one pattern to appear at least once in a sequence is the sum of the probabilities of its first occurrence 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 \( x_\ldots \) denotes \( (x_1 \ldots x_{k-1}) \), the largest prefix of \( x \), for then \( k \leq i \leq \ell \),
\[
Q_x(i) = \sum_{j=|x|-1}^{i-1} Q_{x_1 \ldots x_{k-1}}(j) q_{x_{k-1}}(x_k) \quad (i-j).
\]

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:
\[
M_x = \begin{pmatrix}
Q_x(0) & 0 & \cdots & 0 \\
Q_x(1) & Q_x(0) & \cdots & 0 \\
\vdots & \vdots & \ddots & \vdots \\
0 & 0 & \cdots & Q_x(|x|)
\end{pmatrix}.
\]

Then, \( Q_x \) is obtained as
\[
M_x \times \begin{pmatrix}
q_{x_{k-1}}(1) \\
q_{x_{k-1}}(2) \\
\vdots \\
q_{x_{k-1}}(\ell-|x|+1)
\end{pmatrix}.
\]

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 simplicity, we now consider that all sequences of \( DB \) have the same length \( \ell \). 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 \( \ell \) 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:

\[
\begin{align*}
Z_x^i &= 1 \text{ if the } i^{th} \text{ sequence supports } x, \\
Z_x^i &= 0 \text{ if not,}
\end{align*}
\]

that is, \( Z_x^i = 1 \) with the probability \( p(x) \). Then \( \text{Supp}(x) \) is the sum of those independant identically distributed (i.i.d.) random variables of Bernoulli, and

\[
\text{Supp}(x) \sim \text{Bin}(p(x), n).
\]

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

\[
\Pr\{\text{Bin}(p(x), n) \geq \text{Supp}_{\text{obs}}(x)\}.
\]

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

Let us now describe the algorithm \( \text{Un-SP} \) which extracts unexpected sequential patterns.

Given probabilities of first occurrence, a minimum support \( \min\text{Supp} \), and a threshold \( \varepsilon \) for the p-values, \( \text{Un-SP} \) 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 \( \min\text{Supp} \), 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. \( \text{Un-SP} \) is described in 2. At the end of the algorithm, we call for two functions \( \text{new.pref} \) and \( \text{new.letter} \) which allow to go to the next branch. In summary, \( \text{Un-SP} \) is a classical depth-first algorithm for sequential pattern extraction but outlines only unexpected ones thanks to a filtering step.

4 Experiments

In this section, we describe our experiments on real biological datasets. The algorithm \( \text{Un-SP} \) 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 \( \min\text{Supp} = 90\% \) 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 for \( \varepsilon = 1 \times 10^{-1}, 1 \times 10^{-2}, 1 \times 10^{-3} \). For example, if \( \min\text{Supp} = 80\% \), there are 367696 sequential patterns, but only 2082 of p-value less than \( 1 \times 10^{-1}, 321 \) of p-value less than \( 1 \times 10^{-2}, \) and 57 of p-value less than \( 1 \times 10^{-3} \). 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.

<table>
<thead>
<tr>
<th>Pattern</th>
<th>p-value</th>
</tr>
</thead>
<tbody>
<tr>
<td>CATATAAAACATCA</td>
<td>0.0004</td>
</tr>
<tr>
<td>CATATAAAACACAC</td>
<td>0.0005</td>
</tr>
<tr>
<td>CATTATAAAACAC</td>
<td>0.0002</td>
</tr>
<tr>
<td>CATTATAAAATACA</td>
<td>0.0008</td>
</tr>
<tr>
<td>CTTTATAAAAACAC</td>
<td>0.0004</td>
</tr>
<tr>
<td>CTTTTTTTTTTTTT</td>
<td>0.0007</td>
</tr>
<tr>
<td>CTTTTTTTTTTTTT</td>
<td>0.0001</td>
</tr>
</tbody>
</table>

Table 3: \( \min\text{Supp} = 90\% \) and \( \varepsilon = 1 \times 10^{-2} \)
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 sequences which, roughly, fold in a compact structure and are associated to a biological function. Many proteins include sequences which observed and predicted frequencies are large enough to indicate that they are functional. In [Li et al., 2008a], 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 ATTC TAGC, where the two boxes ATTC and TAGC occur (possibly with errors) with two “do-not-care” symbols in-between. In [Robin et al., 2002], the probability of occurrence 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 occurrence is computed by the inclusion-exclusion principle, but only for structured motifs with a fixed length gap. The probabilities of occurrence 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], subse- quences 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, these 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 occurrences 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 algorithm that identifies 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 raise 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.

Figure 1: Number of sequential patterns extracted.
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 occurrences of items belonging to a same sequential pattern, such as in [Masseglia et al., 2008].

References


