Applied Probability Trust (10 December 2007)

PATTERN MARKOV CHAINS: OPTIMAL MARKOV CHAIN EMBEDDING THROUGH DETERMINISTIC FINITE AUTOMATA

Submitted: June 19, 2006; Revised: September 18, 2007 ; November 28, 2007; Accepted: December 10, 2007

GRÉGORY NUEL,* *University Paris Descartes, MAP5, UMR CNRS 8145*

Abstract

In the framework of pattern in random texts, the Markov chain embedding techniques consist to turn the occurrences of a pattern over an order m Markov sequence into those of a subset of states into an order 1 Markov chain. In this paper, we use the theory of language and automata to provide space optimal Markov chain embedding through the new notion of Pattern Markov Chain (PMC) an give explicit constructive algorithms to build the PMC associated to any given pattern problem. The interest of PMC is then illustrated through the exact computation of p-values which complexity is discussed and compared to other classical asymptotic approximations. Finally, we consider two illustrative examples of highly degenerated pattern problems (structured motifs and PROSITE signature) which further illustrate the usefulness of our approach.

Keywords: language; regular expression; exact distribution; structured motifs; PROSITE signatures

AMS 2000 Subject Classification: Primary 65C40

Secondary

1. Introduction

The theory concerning pattern and motif occurrence in random strings has been of interest since 1950s. Computational molecular biology has been a major area of application of this theory since late 1980s. A variety of methods have been suggested in the literature for treating exact distribution properties associated with pattern occurrence. For example, combinatorial and classical probabilistic methods have been used in Guibas and Odlyzko (1981); Chryssaphinou and Papastavridis (1990); Robin and Daudin (1999, 2001);

* Postal address: 45 rue des Saints-Peres, 75270 Paris Cedex 06, France

Stefanov (2003), Markov chain embeddings - in Fu (1996); Chadjiconstantinidis et al. (2000); Antzoulakos (2001); Fu and Chang (2002), Markov renewal embeddings - in Biggins and Cannings (1987), exponential families with either Markov chain or Markov renewal embeddings - in Stefanov and Pakes (1997, 1999); Stefanov (2000) , and martingale techniques - in Li (1980); Glaz et al. (2006).

An overview of some of these methods has been provided by Reinert et al. (2000). None of the available methods is uniformly superior as far as computation of relevant distributions is concerned. Furthermore, it has been noticed that the computational effort is substantial for any of the available methods when the pattern cardinality (number of string the pattern contains) becomes relatively large.

Inspiring from pattern matching theory, Nicodeme et al. (2002) first proposed to overcome this problem using Deterministic Finite Automata (DFA) in order to get moment generating function of pattern counts through the Chomsky and Schützenberger algorithm. A very similar approach using exponential families have also been proposed by Crochemore and Stefanov (2003).

The purpose of this paper is to push forward the connexion between patterns and automata by introducing an optimal Markov chain embedding through the notion of Pattern Markov Chains (section 2). We then illustrate how this new tool can be used to perform efficient exact and approximate pattern computations (section 3) and the paper ends with two highly degenerated biological patterns applications where our method proves its practical usefulness (section 4).

2. Pattern Markov Chains

2.1. Automata and languages

In this part we first introduce some classical definitions and results of the well known theory of languages and automata Hopcroft et al. (2001).

We consider $\mathcal{A} = \{a_1, \ldots, a_k\}$ a *finite alphabet* which elements are called *letters*. A *word* (or *sequence*) over $\mathcal{A}$ is a sequence of letters and a *language* over $\mathcal{A}$ is a set of words. We denote by ε the *empty word*. For example abbaba is a word over the binary alphabet $\mathcal{A} = \{\mathrm{a}, \mathrm{b}\}$ and $\mathcal{L} = \{\mathrm{ab}, \mathrm{abbaba}, \mathrm{bbbbb}\}$ is a language over $\mathcal{A}$.

The *product* $\mathcal{L}_1 \cdot \mathcal{L}_2$ (the dot could be omitted) of two languages is the language $\{w_1 w_2, w_1 \in \mathcal{L}_1, w_2 \in \mathcal{L}_2\}$ (where $w_1 w_2$ is the concatenation – or product – of w_1 and w_2. If $\mathcal{L}$ is a language, $\mathcal{L}^n = \{w_1 \ldots w_n$ with $w_1, \ldots w_n \in \mathcal{L}\}$ and the *star closure* of $\mathcal{L}$ is defined by $\mathcal{L}^* = \cup_{n \geqslant 0} \mathcal{L}^n$. The language $\mathcal{A}^*$ is hence the set of all possible words over $\mathcal{A}$. For example we have $\{\mathrm{ab}\} \cdot \{\mathrm{abbaba}, \mathrm{bbbbb}\} = \{\mathrm{ababbaba}, \mathrm{abbbbbb}\}$;

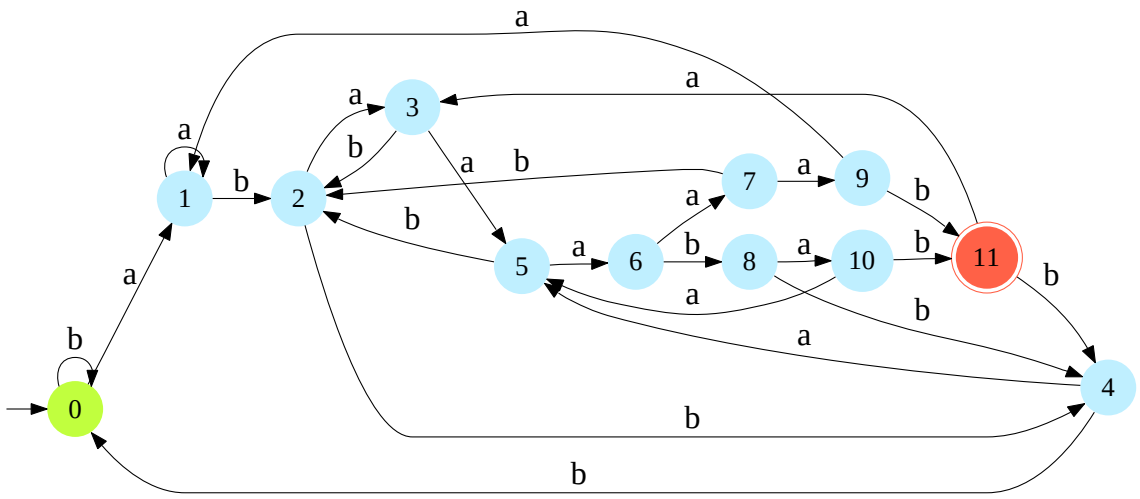

FIGURE 1: Graphical representation of the DFA $(\mathcal{A}, \mathcal{Q}, s, \mathcal{F}, \delta)$ with $\mathcal{A} = \{\mathrm{a}, \mathrm{b}\}$, $\mathcal{Q} = \{0, 1, 2, \ldots, 10, 11\}$, $s = 0$, $\mathcal{F} = \{11\}$ and $\delta(0, \mathrm{a}) = 1$, $\delta(0, \mathrm{b}) = 0$, $\delta(1, \mathrm{a}) = 1$, $\delta(1, \mathrm{b}) = 2$, $\delta(2, \mathrm{a}) = 3$, $\delta(2, \mathrm{b}) = 4$, $\delta(3, \mathrm{a}) = 5$, $\delta(3, \mathrm{b}) = 1$, $\delta(4, \mathrm{a}) = 5$, $\delta(4, \mathrm{b}) = 0$, $\delta(5, \mathrm{a}) = 6$, $\delta(5, \mathrm{b}) = 2$, $\delta(6, \mathrm{a}) = 7$, $\delta(6, \mathrm{b}) = 8$, $\delta(7, \mathrm{a}) = 9$, $\delta(7, \mathrm{b}) = 2$, $\delta(8, \mathrm{a}) = 10$, $\delta(8, \mathrm{b}) = 4$, $\delta(9, \mathrm{a}) = 1$, $\delta(9, \mathrm{b}) = 11$, $\delta(10, \mathrm{a}) = 5$, $\delta(10, \mathrm{b}) = 11$, $\delta(11, \mathrm{a}) = 3$ and $\delta(11, \mathrm{b}) = 4$. This DFA is the smallest one that recognize the language $\mathcal{L} = \mathcal{A}\mathcal{W}_1$ with $\mathcal{A} = \{\mathrm{a}, \mathrm{b}\}$, $\mathcal{W}_1 = \mathrm{ab}\mathcal{A}^1\mathrm{aa}\mathcal{A}^1\mathrm{ab}$ and hence $|\mathcal{W}_1| = 4$.

$\{\mathrm{ab}\}^3 = \{\mathrm{ababab}\}$ and $\{\mathrm{ab}\}^* = \{\varepsilon, \mathrm{ab}, \mathrm{abab}, \ldots\}$

A *regular language* is either the empty word, or a single letter, or obtained by union, product and star closure of regular languages. $\mathcal{A}^*$ is regular. Any finite language is regular.

Definition 1. If $\mathcal{A}$ a finite alphabet, $\mathcal{Q}$ a finite set of states, $s \in \mathcal{Q}$ a starting state, $\mathcal{F} \subset \mathcal{Q}$ a subset of final states and $\delta : \mathcal{Q} \times \mathcal{A} \to \mathcal{Q}$ a transition function then $(\mathcal{A}, \mathcal{Q}, s, \mathcal{F}, \delta)$ is a *Deterministic Finite Automaton* (DFA). For all $a = a_1 \ldots a_{d-1} a_d \in \mathcal{A}^d$ ($d \geqslant 2$) and $q \in \mathcal{Q}$ we recursively define $\delta(q, a_1 \ldots a_{d-1} a_d) = \delta(\delta(q, a_1 \ldots a_{d-1}), a_d)$. A word $w \in \mathcal{A}^h$ is is *accepted* (or *recognized*) by the DFA if $\delta(s, w) \in \mathcal{F}$. The set of all words accepted by a DFA is called its language. See on figure 1 a graphical representation of a DFA.

We can now give the most important result of this part which is a simple application of the classical Kleene and Rabin & Scott theorems Hopcroft et al. (2001):

Theorem 1. *For any rational language $\mathcal{L}$ there exists a unique (up to a unique isomorphism) smallest DFA which language is $\mathcal{L}$.*

k	1	2	3	4	5	6	7	8	9	10	11
$\lvert\mathcal{W}_k\rvert$	4	16	64	256	1 024	4 096	16 384	65 536	262 144	1 048 576	4 194 304
L	12	27	57	122	262	562	1 207	2 592	5 567	11 957	25 682
F	1	3	6	13	28	60	129	277	595	1 278	2 745

TABLE 1: Characteristics of the smallest DFA that recognizes the language $\mathcal{L} = \mathcal{A}\mathcal{W}_k$ with $\mathcal{A} = \{\mathrm{a}, \mathrm{b}\}$ and $\mathcal{W}_k = \mathrm{ab}\mathcal{A}^k\mathrm{aa}\mathcal{A}^k\mathrm{ab}$. The pattern cardinality is $|\mathcal{W}_k| = 2^k \times 2^k = 4^k$, L is the total number of states and F the number of final states.

2.2. Connexion with patterns

We call *pattern* over the finite alphabet $\mathcal{A}$ any finite language over the same alphabet such as no element is included into another one (this last condition is used to simplify many definitions and results by avoiding degenerated cases). For any pattern $\mathcal{W}$ any DFA that recognizes the regular language $\mathcal{A}^*\mathcal{W}$ is said to be *associated* with $\mathcal{W}$. According to theorem 1, there exists a unique (up to unique isomorphism) smallest DFA associated with a given pattern.

For example, if we work with the binary alphabet $\mathcal{A} = \{\mathrm{a}, \mathrm{b}\}$ then the smallest DFA associated with the pattern $\mathcal{W}_1 = \mathrm{ab}\mathcal{A}^1\mathrm{aa}\mathcal{A}^1\mathrm{ab}$ has $L = 12$ states and $F = 1$ final state. A graphical representation of this DFA is given in figure 1.

It is well known from the pattern matching theory Cormen et al. (1990); Crochemore and Hancart (1997) that such a DFA provides a simple way to find all occurrences of the corresponding pattern in a sequence. In the following, we will see how to exploit this remarkable property to study the distribution of patterns.

One should not that in the special case where our pattern contains only one word there is an easy way to build its smallest associated DFA:

Proposition 1. *If $\mathcal{W} = \{w = w_1 \dots w_h\}$ a single word of length h then its smallest associated DFA is of size $L = h + 1$ and defined by $\mathcal{Q} = \{\varepsilon, w_1, w_1w_2, \dots, w\}$ the set of all prefixes of w, $s = \varepsilon$, $\mathcal{F} = \{w\}$ and for all $q \in \mathcal{Q}$ and $a \in \mathcal{A}$, $\delta(q, a)$ is simply defined as the longest suffix of qa (concatenation of q and a) in $\mathcal{Q}$.*

In the case of a general pattern, a similar method can produce an associated DFA (consider for $\mathcal{Q}$ the union of all pattern prefixes) but it would not necessary be the smallest one. In order to be more efficient in the DFA design, one should use instead the classical and well known algorithms provided by the theory of

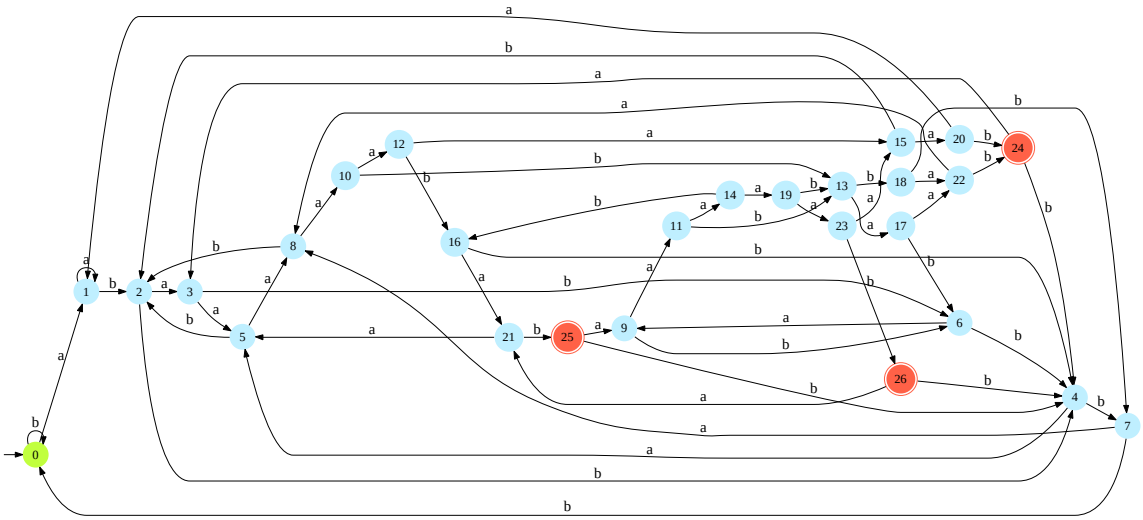

FIGURE 2: Graphical representation of the smallest DFA associated with $\mathcal{W}_2 = \mathrm{ab}\mathcal{A}^2\mathrm{aa}\mathcal{A}^2\mathrm{ab}$. This DFA has $L = 27$ states including $F = 3$ final states.

languages and automata (regular expression to FSA, determinization and epsilon removal).

For example, let us consider the pattern $\mathcal{W}_k = \mathrm{ab}\mathcal{A}^k\mathrm{aa}\mathcal{A}^k\mathrm{ab}$ ($k \geqslant 1$) over the binary alphabet $\mathcal{A} = \{\mathrm{a}, \mathrm{b}\}$. Table 1 shows that the number of final states is (often dramatically) smaller than the cardinal of the pattern. $\mathcal{W}_1$ is recognized by the DFA of figure 1, $\mathcal{W}_2$ by the one of figure 2 and $\mathcal{W}_{11}$, a pattern with a cardinal of several millions, is recognized by a DFA having only a few thousands states.

Assuming from now that a DFA (smallest or not) associated with our pattern has been built, we can give the main result of this part:

Theorem 2. *if $X = X_1X_2 \dots X_i \dots$ is a i.i.d. sequence on $\mathcal{A}$, $\mathcal{W}$ a pattern and $(\mathcal{A}, \mathcal{Q}, s, \mathcal{F}, \delta)$ an associated DFA then sequence $Y = Y_0Y_1Y_2 \dots Y_i$ defined by*

$$Y_0 = s \quad \textit{and} \quad Y_i = \delta(Y_{i-1}, X_i) \quad \textit{for all} \quad i \geqslant 1$$

is an order 1 *Markov chain which transition matrix is given by*

$$\Pi(p,q) = \begin{cases} \mathbb{P}(X_1 = a) & \textit{if } \delta(p,a) = q \\ 0 & \textit{if } q \notin \delta(p, \mathcal{A}) \end{cases}$$

and such as occurrences of $\mathcal{W}$ in X correspond to occurrences of a subset of letters in Y (here $\mathcal{F}$). A Markov chain having these properties is called a Pattern Markov Chain *(PMC). Moreover, if the DFA is optimal (i.e.*

has the smallest number of states) then the resulting PMC has the same property.

Proof. By definition, the sequence Y is obviously an order 1 Markov chain. Moreover, if an occurrence of $\mathcal{W}$ ends at position i in X, the sequence $X_1 \dots X_i$ ends with an occurrence of the pattern and is therefore an element of $\mathcal{A}^*\mathcal{W}$ and thus is accepted by the DFA which means that $Y_i \in \mathcal{F}$ and the first part of the theorem is proved.

Let us now assume that it exists a set $\mathcal{Q}$, a subset $\mathcal{F} \subset \mathcal{Q}$ and a function $G : \mathcal{A}^* \to \mathcal{Q}^*$ such as:

i) $\forall x \in \mathcal{A}^*$ we denote $y = G(x)$; $\forall 0 \leqslant i \leqslant |x|$, $\mathcal{W}$ ends in position i in $x \iff y_i \in \mathcal{F}$;

ii) if X is i.i.d. then $Y = G(X)$ is an order 1 Markov chain.

For all $x \in \mathcal{A}^*$ and $a \in \mathcal{A}$ we denote by $\Delta(x, a)$ the state in position $|xa|$ in $f(xa)$ and we define recursively the function $\widetilde{G} : \mathcal{A}^* \to \mathcal{Q}^*$ by $\widetilde{G}(\varepsilon) = G(\varepsilon)$ and $\widetilde{G}(xa) = \widetilde{G}(x)\Delta(x, a) \quad \forall a \in \mathcal{A}, x \in \mathcal{A}^*$. We define now $\widetilde{\Delta}(\widetilde{G}(x), a) = \Delta(x, a)$ on the quotient space $(\mathcal{A}^*)_{\mathcal{R}}$ where $x\mathcal{R}x' \iff \widetilde{G}(x) = \widetilde{G}(x')$.

Thanks to (ii), it exists $\delta : \mathcal{Q} \times \mathcal{A} \to \mathcal{Q}$ such as $\widetilde{\Delta}(yq, a) = \delta(q, a)$ for all $yq \in \widetilde{G}(\mathcal{A}^*)$ and $a \in \mathcal{A}$. Hence $(\mathcal{A}, \mathcal{Q}, s = \widetilde{G}(\varepsilon)_0, \mathcal{F}, \delta)$ is a DFA associated with $\mathcal{W}$ and the second part of the theorem is proved.

One should note that the transition matrix of a PMC is sparse (only $k \times L$ non zero terms among L^2, where k is the alphabet size) and that we have a natural decomposition of this transition matrix into $\Pi = P + Q$ where Q contains all transitions toward counting states and P the regular ones.

Example 1. Let us consider the pattern $\mathcal{W}_1 = \mathrm{ab}\mathcal{A}^1\mathrm{aa}\mathcal{A}^1\mathrm{ab}$ over the binary alphabet $\mathcal{A} = \{\mathrm{a}, \mathrm{b}\}$. Its smallest associated DFA is represented on figure 1. If X is the original sequence, we build the PMC Y as follows (final states in bold):

$X =$	$-$	a	b	a	a	a	b	b	a	a	a	a	b	b	a	a	b	a	b	a	b
$Y =$	0	1	2	3	5	6	8	4	5	6	7	9	**11**	4	5	6	8	10	**11**	3	2

We see two occurrences of $\mathcal{W}_1$: one ending in position 12 (abbaaaab) and one in position 18 (abbaabab,

overlapping the previous occurrence). The transition matrix of Y is given by

$$\Pi = \begin{pmatrix}
\mu_{\mathrm{b}} & \mu_{\mathrm{a}} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & \mu_{\mathrm{a}} & \mu_{\mathrm{b}} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & \mu_{\mathrm{a}} & \mu_{\mathrm{b}} & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & \mu_{\mathrm{b}} & 0 & \mu_{\mathrm{a}} & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
\mu_{\mathrm{b}} & 0 & 0 & 0 & 0 & \mu_{\mathrm{a}} & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & \mu_{\mathrm{b}} & 0 & 0 & 0 & \mu_{\mathrm{a}} & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & \mu_{\mathrm{a}} & \mu_{\mathrm{b}} & 0 & 0 & 0 \\
0 & 0 & \mu_{\mathrm{b}} & 0 & 0 & 0 & 0 & 0 & 0 & \mu_{\mathrm{a}} & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & \mu_{\mathrm{b}} & 0 & 0 & 0 & 0 & 0 & \mu_{\mathrm{a}} & 0 \\
0 & \mu_{\mathrm{a}} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \mu_{\mathrm{b}}^{*} \\
0 & 0 & 0 & 0 & 0 & \mu_{\mathrm{a}} & 0 & 0 & 0 & 0 & 0 & \mu_{\mathrm{b}}^{*} \\
0 & 0 & 0 & \mu_{\mathrm{a}} & \mu_{\mathrm{b}} & 0 & 0 & 0 & 0 & 0 & 0 & 0
\end{pmatrix}$$

where transitions with * belong to Q and with $\mu_{\cdot} = \mathbb{P}(X_1 = \cdot)$.

As explained in the introduction, the authors of Nicodeme et al. (2002) proposed to use pattern's DFA to get the pattern generating function through the Chomsky & Schützenberger algorithm and derive from it exact results and asymptotic moments. More recently, Crochemore and Stefanov (2003) used the pattern's automaton conjointly with exponential families results in the same aim. Instead of focusing of generating function only (as done in these papers), we propose here a more straightforward and practical approach consisting to exploit our new PMC to improve a wide range of classical pattern methods.

2.3. Extensions

The methods we have presented until now are only valid for overlapping occurrences of a pattern in a i.i.d. sequence. We propose here to extend our results to Markov sequences or to renewal occurrences.

2.3.1. *Markov chains* In order to extend our results to Markov chain sequences, we first need to introduce the following definition

Definition 2. A DFA $(\mathcal{A}, \mathcal{Q}, \mathcal{F}, s, \delta)$ where it exists $q \in \mathcal{Q}$ and $a, b \in \mathcal{A}^m$ such as $a \neq b$ and $\delta(q, a) = \delta(q, b)$ is called m-*ambiguous*. A DFA which is not m-ambiguous is also called m-*unambiguous*.

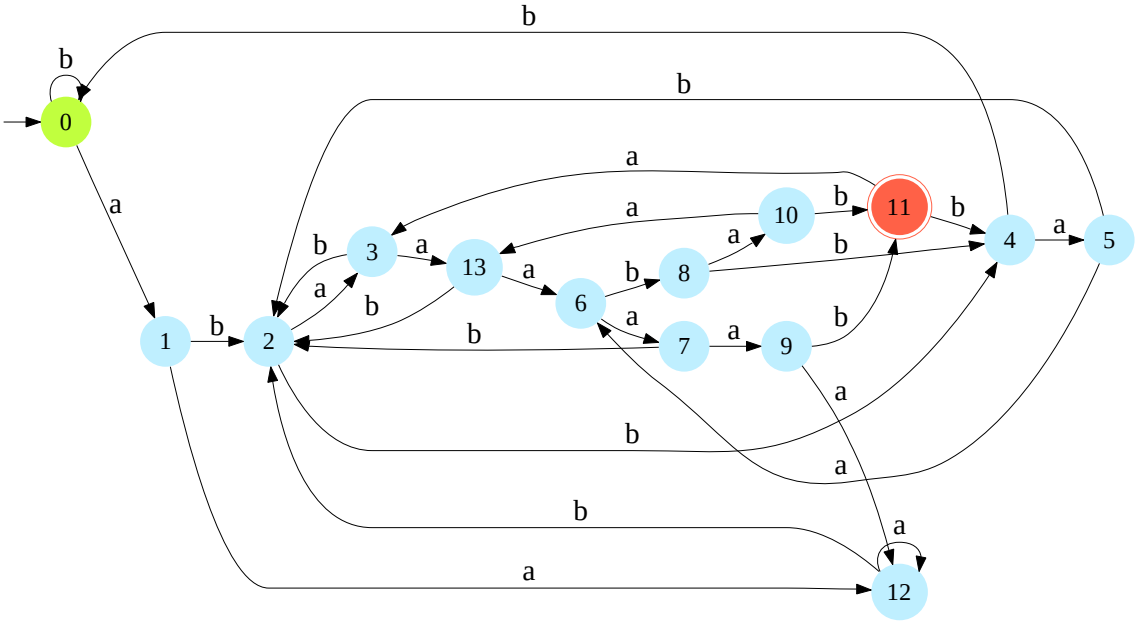

FIGURE 3: Graphical representation of the smallest non 2-ambiguous DFA associated with $\mathcal{W}_1 = \mathrm{ab}\mathcal{A}^1\mathrm{aa}\mathcal{A}^1\mathrm{ab}$ with $\mathcal{A} = \{\mathrm{a}, \mathrm{b}\}$. This DFA have been built from the DFA of figure 1 through algorithm 1.

Please note that the m-ambiguity presented here is different from the classical notion of *ambiguity* for DFA (meaning that it exists two different path to recognize the same language element).

For any DFA $(\mathcal{A}, \mathcal{Q}, s, \mathcal{F}, \delta)$ we define for all $q \in \mathcal{Q}$ and for all $m \geqslant 1$ the following notations:

$$\delta^{-m}(q) = \{a \in \mathcal{A}^m, \exists p \in \mathcal{Q}, \delta(p,a) = q\} \quad \text{and} \quad \Delta^{-1}(q) = \{p \in \mathcal{Q}, \exists a \in \mathcal{A}, \delta(p,a) = q\}$$

Hence, such a DFA is m-unambiguous if all $\delta^{-m}(q)$ are singletons.

Theorem 3. *if $X = X_1 \dots X_n$ is an order $m \geqslant 1$ Markov sequence on $\mathcal{A}$, $\mathcal{W}$ a pattern and $(\mathcal{A}, \mathcal{Q}, s, \mathcal{F}, \delta)$ a non m-ambiguous DFA which language is $\mathcal{A}^*\mathcal{W}$ then the sequence $Y = Y_m \dots Y_n$ defined by*

$$Y_0 = s \quad \textit{and} \quad Y_i = \delta(Y_{i-1}, X_i) \quad \textit{for all} \quad 1 \leqslant i \leqslant n$$

is an order 1 Markov chain which transition matrix is given by

$$\Pi(p,q) = \begin{cases} \mathbb{P}(X_{m+1} = b | X_1 \dots X_m = \delta^{-m}(p)) & \text{if } \delta(p,b) = q \\ 0 & \text{if } q \notin \delta(p, \mathcal{A}) \end{cases}$$

and such as occurrences of $\mathcal{W}$ in X correspond to occurrences of a subset of letters in Y. Y is therefore a PMC.

Proof. The proof is very similar to the one of the i.i.d. case except that the non m-ambiguity is obviously required to insure that all $\delta^{-m}(p)$ are singletons.

Using this theorem, it is possible to apply all preceding methods to Markovian sequence. But the key question is of course: is it possible to build a non m-ambiguous pattern DFA and how ?

In Nicodeme et al. (2002), the authors explain (algorithm 6) that this can be done starting from a DFA associated with the pattern by duplicating states until all ambiguities have been removed. This, of course, is exactly what we need to do. However in this paper, we want to propose a more explicit approach with algorithm 1.

As suggested by Nicodeme et al. (2002), this algorithm simply duplicates state for which it exits a m-ambiguity while preserving the DFA ability to recognize its language. As only the necessary states are duplicated, this algorithm also preserves the optimality of produced DFA.

Require: $A = (\mathcal{A}, \mathcal{Q}, s, \mathcal{F}, \delta)$ is a $(m-1)$-unambiguous DFA that recognize $\mathcal{W}$

INITIALIZATION:

$\mathcal{Q}_0 = \mathcal{Q}$, $\forall q \in \mathcal{Q}$, $\mathcal{D}_q = \delta^{-m}(q)$ and $\mathcal{G}_q = \Delta^{-1}(q)$

MAIN LOOP:

for all $q \in \mathcal{Q}_0$ **do**

  while $|\mathcal{D}_q| > 1$ **do**

    take $a = a_1 \dots a_m \in \mathcal{D}_q$

    add a new state q_a to $\mathcal{Q}$

    if $q \in \mathcal{F}$ then add q_a to $\mathcal{F}$

    define $\mathcal{D}_{q_a} = \{a\}$ and $\mathcal{G}_{q_a} = \emptyset$

    for all $b \in \mathcal{A}$ do $\delta(q_a, b) = \delta(q, b)$ and add q_a to $\mathcal{G}_{\delta(q,b)}$

    for all $p \in \mathcal{G}_q$

    if $\delta(p, a_m) = q$ and $\delta^{-(m-1)}(p) = a_1 \dots a_{m-1}$ (empty condition if $m = 1$) **then**

      $\delta(p, a_m) = q_a$ and add p to $\mathcal{G}_{q_a}$

    for all $p \in \mathcal{G}_q$, if $q \notin \delta(p, \mathcal{A})$ then remove q from $\mathcal{G}_q$

    remove a from $\mathcal{D}_q$

Algorithm 1: Build a m-unambiguous DFA that recognize $\mathcal{W}$ from a $(m-1)$-unambiguous DFA (empty condition if $m = 1$) having the same property. Let us note that we still have $\mathcal{D}_q = \delta^{-m}(q)$ and $\mathcal{G}_q = \Delta^{-1}(q)$ at the end of algorithm.

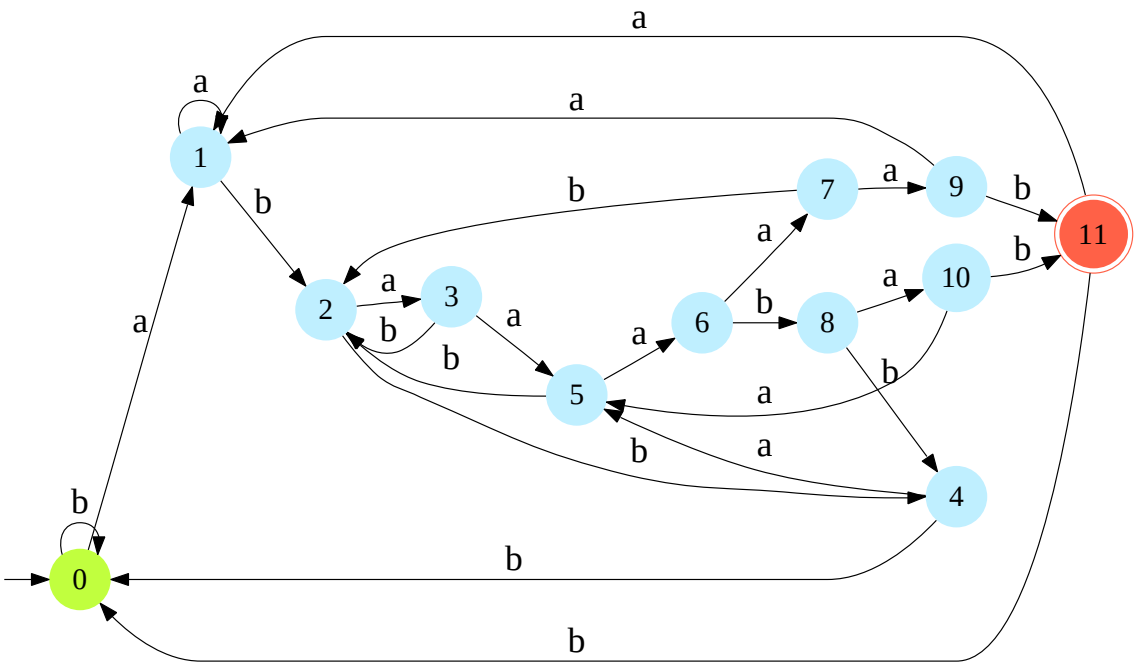

FIGURE 4: Graphical representation of a renewal DFA associated with $\mathcal{W}_1 = \mathrm{ab}\mathcal{A}^1\mathrm{aa}\mathcal{A}^1\mathrm{ab}$ with $\mathcal{A} = \{\mathrm{a}, \mathrm{b}\}$. This DFA have been built from the DFA of figure 1 through proposition 2.

In order to achieve non m-ambiguity one could hence successively remove 1-ambiguity, then 2-ambiguity and so on till we finally remove m-ambiguity having used a total of m applications of the algorithm 1. For example, we can use this approach to transform the 2-ambiguous DFA of figure 4 ($\delta^{-2}(1) = \{\mathrm{aa}, \mathrm{ba}\}$) into the non 2-ambiguous one of figure 3 by duplicating the state 1 into states 1 and 12.

2.3.2. *Renewal occurrences* We first recall that a renewal occurrence (also called non-overlap occurrences) of a given pattern is an occurrence which does not overlap any previously counted occurrence. For example: $X = \mathrm{abababbaba}$ contains three overlapping occurrences of aba but only two renewal ones (as the second occurrence overlaps the first one).

Adapting pattern methods to such kind of occurrences usually requires a lot of work, but with our approach (as already pointed by Nicodeme et al., 2002), we only need a small modification of our DFA:

Proposition 2. *If* $(\mathcal{A}, \mathcal{Q}, s, \mathcal{F}, \delta)$ *is a DFA which accepts* $\mathcal{L} = \mathcal{A}^*\mathcal{W}$ *then*

$$\delta(f, a) = \delta(s, a) \quad \forall f \in \mathcal{F} \quad \textit{and} \quad \forall a \in \mathcal{A}$$

will transform the DFA to accept only the texts ending with a renewal (i.e. non overlapping) occurrence of $\mathcal{W}$.

Proof. This is trivial since restarting the DFA from s after each occurrence means that past is not taken into account.

Once this transformation has been done, all previous results will hold for renewal occurrences using our modified DFA. One should note that when doing so, the pattern self-overlapping matrix is obviously null and hence makes compound Poisson approximations easier to use as they are only simple Poisson approximations.

One can also extend the notion of renewal occurrences to the one of d*-renewal* occurrences for which we have to wait d steps after a given occurrence to accept another one (thus, renewal occurrences and 0-renewal ones are exactly the same). In order to consider d-renewal occurrences of a pattern $\mathcal{W}$ we simply need to count renewal occurrences of $\mathcal{W}\mathcal{A}^d$.

3. Using PMC

3.1. Exact distribution

DFA have been used by Nicodeme et al. (2002) and Crochemore and Stefanov (2003) to obtain moment-generating functions of the number of occurrence of any pattern in a random sequence. With the help of efficient numerical algorithms (*e. g.* fast Taylor expansion), it is hence possible compute moments or p-values. However, the computational cost of the generating function itself could be important and, as a consequence, more straightforward approaches (like direct moment computations) are often more efficient.

In this part, we consider precisely such a more direct approach by showing how we can use PMC to compute efficiently exact p-values. Our approach consists first to produce through PMC an optimal Markov chain embedding of the problem and then to use recurrence relation exploiting the sparse structure of the transition matrix to perform the computations.

The technique of Markov chain embedding (also called finite Markov chain imbedding – FMCI) have been used for pattern problem by many authors Fu and Koutras (1994); Lou (1996); Fu and Lou (2003). If many embedded Markov chain can be build for a given problem, the design of a space efficient one is of course of critical interest for practical applications. We propose here to solve this problem by showing the very simple connexion that exists between PMC and FMCI.

Let $\mathcal{W}$ be a pattern and $(\mathcal{A}, \mathcal{Q}, s, \mathcal{F}, \delta)$ an associated (smallest or not) DFA. We denote by Y the corresponding PMC which transition matrix is denoted $\Pi = P + Q$ where Q contains all transitions toward final states and P the regular ones.

Definition 3. For any $c \in \mathbb{N}$ we define the FMCI Z by

$$Z_j = \begin{cases} (Y_j, N_j) & \text{if } N_j < c \\ f & \text{if } N_j \geqslant c \end{cases}$$

where N_j is the number of pattern occurrences in $X_1 \dots X_j$.

Proposition 3. *Ordering the $cL+1$ of states of Z as $\{(1,0), \dots, (L,0), (1,1), \dots, (L,1), \dots, (1,c-1), \dots, (L,c-1), f\}$, the corresponding transition matrix is given by*

$$\Pi = \left(\begin{array}{c|c} R & v \\ \hline 0 & 1 \end{array}\right)$$

where R (dimension $cL \times cL$) and v (dimensions $cL \times 1$) are defined by blocks of size L:

$$R_{i,j} = \begin{cases} P & \text{if } i = j \\ Q & \text{if } i+1 = j \\ 0 & else \end{cases} \qquad \text{and} \quad v_i \equiv 0 \text{ for } 1 \leqslant i < c \text{ and } v_c = \Sigma_Q$$

where Σ_Q is the column vector resulting of the sum of Q.

Proof. Obvious since transitions in P will not increment the number of occurrences while transitions in Q will increment it by one.

Example 2. For example if $c = 2$ we get the following transition matrix:

$$\Pi = \left(\begin{array}{cc|c} P & Q & 0 \\ 0 & P & \Sigma_Q \\ \hline 0 & 0 & 1 \end{array}\right)$$

As proposed in Nuel (2006a) it is hence possible to get the p-values we are looking for, through efficient recurrence relations:

Theorem 4. *For all $n \geqslant 1$ and $1 \leqslant i \leqslant k$ we have*

$$\mathbb{P}(N_n < c | X_1 = i) = \left(u^{n-1}\right)_i \quad \text{and} \quad \mathbb{P}(N_n \geqslant c | X_1 = i) = \sum_{j=0}^{n-2} \left(v^j\right)_i$$

method	memory complexity	time complexity
exact	$k \times L + N_{\text{obs}} \times L$	$k \times L \times N_{\text{obs}} \times n$
Gaussian	$k \times L + F \times L$	$k \times L + F \times L \times \log n + F^2$
binomial/Poisson	$k \times L$	$k \times L + F + \log N_{\text{obs}}$
geometric Poisson	$k \times L + F^2$	$k \times L + F^2 + N_{\text{obs}}$
compound Poisson	$k \times L + F^2 + N_{\text{obs}}$	$k \times L + F^2 + N_{\text{obs}}^2$
large deviations	$k \times L$	$k \times L$

TABLE 2: Order of magnitude of memory and time complexities for the different statistical approaches. k is the alphabet size, L is the number of states of the associated DFA, F the number of final states, n the sequence length and N_{obs} the observed number of occurrences.

where $(\,)_i$ *denotes the* i^{th} *component of a vector, where for* $x = u$ *or* v *we have* $\forall j \geqslant 0$ *the following size* L *block decomposition:* $x^j = \left(x^j_{(c-1)}, \ldots, x^j_0\right)'$ *and we have the recurrence relations:*

$$x_0^{j+1} = P x_0^j \quad \textit{and} \quad \forall i \geqslant 1 \quad x_i^{j+1} = P x_i^j + Q x_{i-1}^j$$

with $u^0 = (1 \ldots 1)'$ *and* $v^0 = v$.

3.2. Asymptotic approximations

Thanks to Markov embedding, it is possible to obtain very efficiently the exact distribution of a pattern count. However, the complexity involved in this computation is linear with the sequence length n and the number N_{obs} of observed occurrences (see table 2). In many practical situations, this complexity cost may be prohibitive thus justifying the development of faster approximations. A review of such approximations and the practical means to their efficient implementation is proposed in Nuel (2006c).

Table 2 summarize the time and memory complexities for all these approaches. Let us fist point out the alphabet size k and the cardinal L of the PMC state space are critical parameters for all the method since $k \times L$, the number of non-zero terms in the transition matrix of the PMC, is the complexity of a sparse product of this matrix with a vector.

Unlike with the exact approach we have to assume both homogeneity and ergodicity of the underlying sequence Markov model in order to get these approximations. It is then possible to computing exact first and second order moments of the pattern count with a constant complexity with N_{obs} and only a logarithm complexity with n thus resulting in a dramatically improvement over the Markov embedding approach. One should however note that the number F of final states appears both in memory and time complexity in a

linear or quadratic form.

As Binomial and Poisson approximations only require first order moments, the resulting complexities of both these methods are even reduced. The length n of the sequence completely vanishes from the time complexity. Thanks to incomplete beta (binomial) or incomplete gamma (Poisson) functions, it hence possible to compute approximate p-values with a $\log(N_{\text{obs}})$ complexity.

If we turn now to compound Poisson approximations, the complexity $O(F^2)$ both in time and space is required to study the overlapping structure of the pattern. In general, the resulting computation of p-values then require a quadratic complexity with N_{obs} (which can be a prohibitive cost for frequent patterns) but in the particular case when the compound Poisson is reduced to a simple geometric Poisson the complexity is only linear with N_{obs} thanks to the recurrence formulas given in Nuel (2006b).

Finally, large deviation approximations display the smallest complexities as then only rely on sparse products to solve eigen problems related to the transition matrix of the PMC (which can be done efficiently with Arnoldi class algorithm, see Lehoucq et al., 1998). It is however necessary to emphasize that in practice, the large deviations approaches are slower than other approximations (but also more reliable for exceptional patterns).

4. Applications

We propose in this part to illustrate the interest of PMC through two examples of highly degenerated biological patterns.

4.1. Structured motifs

We consider here an important class of DNA patterns (*i. e.* over the alphabet $\mathcal{A} = \{\mathtt{a}, \mathtt{c}, \mathtt{g}, \mathtt{t}\}$) occurring in the regulatory regions of genes (Marsan and Sagot, 2000). These patterns consist in a sequence of two or more strings each occurrences of which are separated by a specific number of letters. For example, the structured pattern $\mathtt{ttgaca}\mathcal{A}^{16:18}\mathtt{tataata}$ is composed by two strings separated by at least 16 and at most 18 letters.

Robin et al. (2002) gave first a Poisson approximation to the problem, and more recently, Stefanov et al. (2006) proposed exact methods to compute the exact distribution of this kind of patterns. In order to demonstrate the efficiency of our new PMC approach, we consider here the same dataset used in both (kindly provided by the authors).

This dataset is composed of a set of 131 sequence of length 100 located in the upstream region of 131

genes of the bacterium *B. Subtilis*. We also consider a set of 71 structured motifs which are good promoter candidates. These motifs are all of the form $w_1\mathcal{A}^{d_1:d_2}w_2$ where w_1, w_2 are two strings and $d_1 \leqslant d_2$ two integers.

For technical considerations, Stefanov et al. (2006) exclude occurrences of the structured motif where w_1 of w_2 occur more than once (for example in segment $\mathcal{A}^{d_1:d_2}$). As explained by the authors, this slightly differs from the usual definition but the two countings (either usual structured motifs or restricted ones) are obviously closely related.

Assuming that the 131 (the number of 130 sequences should have been misspelled in Stefanov et al. (2006) as the dataset contains indeed 131 sequences and as the authors use then subsequently this latter value for all their binomial computations) sequences are generated according to an homogeneous Markov model which parameter are estimated on the dataset, we consider the random variables $(N_i)_{1\leqslant i\leqslant 131}$ (resp. N_i') count the number of occurrences of the pattern (resp. restricted pattern defined above) in the i^{th} sequence. We hence consider $N = \sum_{i=1}^{131} N_i$ and $M = \sum_{i=1}^{131} \mathbb{I}_{N_i\geqslant 1}$ (as well as there restricted versions N' and M').

The table 3 list the 15 most significant structured motifs among the 71 that have been tested. The column $\mathbb{P}_s(M' \geqslant \text{obs})$ is exactly the last column of table 5 in Stefanov et al. (2006) except for two structured motifs which number of occurrences have been somehow miscounted by the authors (`ttgaca`$\mathcal{A}^{16:18}$`atataat` – resp. `gttgaca`$\mathcal{A}^{16:18}$`tataata` – appears in the sequences rpmH, TrnS and veG – resp. rpmH and f82129 – but is only observed twice – resp. once – according to Stefanov et al., 2006).

As M and M' are different countings, this is not a surprise to see differences between columns 4 and 5 of table 3, but as expected, these differences are small.

Our new method also allows us to consider the sum of counts N rather that the number of sequences M where the motif is present. In the particular case of the patterns considered in our example, there is not much differences between the two statistics. However, differences should be more important when considering either smaller patterns or longer sequences. For example, the pattern $\mathcal{W} =$ `atat` appear in 88 sequences of the dataset but its total number of occurrences is 111; the corresponding p-values are $\mathbb{P}(M \geqslant 88) = 1.66 \times 10^{-2}$ and $\mathbb{P}(N \geqslant 111) = 3.50 \times 10^{-4}$.

Even if the cardinality of each of these structured motifs $|\mathcal{W}| = 4^{16} + 4^{17} + 4^{18} = 90\,194\,313\,216 \simeq 9 \times 10^{10}$ is huge, we can see that the size of the smallest associated DFA is far smaller with an order of magnitude of a thousand. This of course allows our PMC approach to be very efficient both in term of memory usage and running time. For example, computing the 71 p-values of the type $\mathbb{P}_s(M \geqslant \text{obs})$ require a total of 25 seconds on a Intel 2.6 GHz P4 workstation while the computations of $\mathbb{P}_s(M' \geqslant \text{obs})$ with the

$\mathcal{W}$	$L(F)$	obs	$\mathbb{P}_s(M' \geqslant \text{obs})$	$\mathbb{P}_s(M \geqslant \text{obs})$	$\mathbb{P}(M \geqslant \text{obs})$	$\mathbb{P}(N \geqslant \text{obs})$
ttgactt$\mathcal{A}^{16:18}$ataataa	$2571(80)$	3	5.77×10^{-10}	7.10×10^{-10}	7.08×10^{-10}	7.53×10^{-10}
ttgaca$\mathcal{A}^{16:18}$atataat	$1527(55)$	3	na	9.45×10^{-9}	9.43×10^{-6}	9.60×10^{-9}
tgactt$\mathcal{A}^{16:18}$ataataa	$2386(80)$	3	1.00×10^{-8}	1.29×10^{-8}	1.29×10^{-8}	1.33×10^{-8}
gttgaca$\mathcal{A}^{16:18}$tataata	$1014(28)$	2	na	1.50×10^{-7}	1.50×10^{-7}	1.51×10^{-7}
ttgactt$\mathcal{A}^{16:18}$atactaa	$2551(60)$	2	1.37×10^{-7}	1.52×10^{-7}	1.52×10^{-7}	1.53×10^{-7}
tgactt$\mathcal{A}^{16:18}$atactaa	$2366(60)$	2	9.18×10^{-7}	1.05×10^{-6}	1.05×10^{-6}	1.06×10^{-6}
ttgaca$\mathcal{A}^{16:18}$tataatg	$1399(34)$	2	2.18×10^{-6}	2.50×10^{-6}	2.50×10^{-6}	2.51×10^{-6}
ttgaca$\mathcal{A}^{16:18}$tatatta	$1435(43)$	2	4.75×10^{-6}	5.48×10^{-6}	5.47×10^{-6}	5.50×10^{-6}
ttgact$\mathcal{A}^{16:18}$tatact	$2537(106)$	2	4.81×10^{-6}	5.71×10^{-6}	5.71×10^{-6}	5.75×10^{-6}
ttgaca$\mathcal{A}^{16:18}$tataata	$1408(43)$	2	5.23×10^{-6}	6.93×10^{-6}	6.92×10^{-6}	7.02×10^{-6}
tgacttt$\mathcal{A}^{16:18}$taataa	$1505(55)$	2	1.12×10^{-5}	1.30×10^{-5}	1.30×10^{-5}	1.30×10^{-5}
gacttt$\mathcal{A}^{16:18}$taataa	$1386(55)$	2	9.52×10^{-5}	1.08×10^{-4}	1.08×10^{-4}	1.08×10^{-4}
gttgaca$\mathcal{A}^{16:18}$atataat	$1066(35)$	1	5.63×10^{-4}	6.10×10^{-4}	6.10×10^{-4}	6.10×10^{-4}
ttgacac$\mathcal{A}^{16:18}$ataataa	$979(28)$	1	6.39×10^{-4}	6.99×10^{-4}	6.98×10^{-4}	6.98×10^{-4}
gttgac$\mathcal{A}^{16:18}$ctataat	$1392(43)$	1	6.39×10^{-4}	6.84×10^{-4}	6.84×10^{-4}	6.84×10^{-4}

TABLE 3: The 15 most significant structured motifs. $\mathcal{W}$ indicates the motif, L (resp. F) the number of states (resp. final states) of the smallest non 1-ambiguous associated DFA, obs is the number of observed occurrences in the dataset and the subscript s means that the probability is computed assuming stationarity.

previous method took 3277 seconds on a IBM F80 computer. Our approach is hence more than 100 times faster than the previous one which is a dramatic improvement.

It is nevertheless important to point out that the computations performed in Stefanov et al. (2006) were not seeking for numerical performance. Moreover, Stefanov et al. (2006) consider the problem as two competing patterns rather than a single (highly degenerated one) which results in a marginal increasement of complexity with the gap length while the single pattern approach presented here is geometrically dependant with this parameter.

On should note that is it possible to adapt the PMC framework to a competing pattern problem by splitting the subset of final states into $\mathcal{F} = \mathcal{F}_1 \cup \mathcal{F}_2$ where $\mathcal{F}_1$ (resp. $\mathcal{F}_2$) contains the final states associated to the pattern w_1 (resp. w_2). If we consider then the corresponding decomposition of the transition matrix $\Pi = P+Q_1+Q_2$ it is then possible to get the the distribution of a structured pattern in a very straightforward

way:

$$\mathbb{P}(w_1\mathcal{A}^d w_2 \text{ starts in } i) = \underbrace{\mu_m P^{i-m}}_{\text{up to } i} \times \underbrace{P^{|w_1|-2}Q_1}_{w_1} \times \underbrace{P^d}_{\text{gap}} \underbrace{\times P^{|w_2|-1}Q_2}_{w_2} \times e_{\mathcal{F}_2}^T$$

If we consider for example $w_1 = \texttt{ttgaca}$, $w_2 = \texttt{atataat}$ and $16 \leqslant d \leqslant 18$, the smallest 1-unambiguous DFA that allows to count both w_1 and w_2 has $L = 16$ states (while the DFA associated to the full structured motif has $L = 1527$ states) we get

$$\mathbb{P}(w_1\mathcal{A}^{16:18}w_2) \simeq \sum_{d=16}^{17}\sum_{i=1}^{100} \mathbb{P}(w_1\mathcal{A}^d w_2 \text{ starts in } i) = 3.06 \times 10^{-5}$$

which is very close to the exact solution (3.02×10^{-5} in Stefanov et al., 2006) despite the fact that important dependencies are not here taken into account.

This alternative approach obviously need more work to deal rigorously with the problem but seems already appealing since it combines the interest of the existing method and of the new one. Indeed most of the complex combinatorial aspects of the problem are embedded in the PMC (which state space is greatly reduced) and, like in Stefanov et al. (2006), dealing with larger gaps is not a problem.

Finally, let us add that our PMC approach to structured motifs have several natural extensions which are likely to be difficult to get with previous approaches:

- structured motifs with degenerated patterns (possibly of variable lengths) instead of simple words;
- structured motifs with more than two patterns;
- heterogeneous background models.

In order to illustrate this last point, we propose to consider the following heterogeneous Markov model over $\mathcal{A} = \{\texttt{a}, \texttt{c}, \texttt{g}, \texttt{t}\}$: the starting distribution μ_1 (MLE estimate using the dataset) is given by:

$$\mu_1 = \left(\frac{50}{131} \quad \frac{17}{131} \quad \frac{23}{131} \quad \frac{41}{131}\right)$$

and the heterogeneous (and arbitrary) transition matrix by:

$$\pi_i(a,b) = \mathbb{P}(X_i = b | X_{i-1} = a) = \frac{(100-i)}{98}\pi^0 + \frac{(i-2)}{98}\pi^1 \quad \forall a,b \in \mathcal{A}, \forall 2 \leqslant i \leqslant 100$$

where

$$\pi^0 = \begin{pmatrix} 0.5 & 0.1 & 0.1 & 0.3 \\ 0.2 & 0.2 & 0.3 & 0.4 \\ 0.4 & 0.3 & 0.2 & 0.1 \\ 0.3 & 0.2 & 0.3 & 0.2 \end{pmatrix} \quad \text{and} \quad \pi^1 = \begin{pmatrix} 0.1 & 0.4 & 0.4 & 0.1 \\ 0.4 & 0.3 & 0.1 & 0.2 \\ 0.6 & 0.2 & 0.1 & 0.1 \\ 0.3 & 0.2 & 0.1 & 0.4 \end{pmatrix}.$$

m	0	1	2	3
L	329	1 393	10 688	134 746
F	30	78	633	3 045

TABLE 4: Characteristics of the smallest non m-ambiguous DFA associated to the cyclic nucleotide-binding domain signature 2 (PS00889): [LIVMF]-G-E-x-[GAS]-[LIVM]-x(5,11)-R-[STAQ]-A-x-[LIVMA]-x-[STACV] (cardinality $\simeq 10^{22}$). L denotes the number of states and F the number of final states.

Using the PMC framework, it is then easy to compute the exact probability to observe at least one occurrence of a structured pattern in a random sequence drawn either according to an homogeneous model or according to the heterogeneous one defined above:

$$\mathbb{P}(N(\texttt{ttgacttA}^{16:18}\texttt{ataataa}) \geqslant 1) = \begin{cases} 6.863712 \times 10^{-6} & \text{with the homogeneous transitions } \pi^0 \\ 8.795492 \times 10^{-8} & \text{with the homogeneous transitions } \pi^1 \\ 1.549870 \times 10^{-6} & \text{with the heterogeneous transitions} \end{cases}$$

4.2. PROSITE signatures

Another interesting family of biological patterns are the signatures of the PROSITE database Hulo et al. (2006). This database contains protein consensus patterns for many of functional families. As protein are simple sequences of amino-acids (size $k = 20$ alphabet), the PROSITE signatures are often highly degenerated.

For example, the cyclic nucleotide-binding domain signature 2 (PS00889 entry of the PROSITE database) is: [LIVMF]-G-E-x-[GAS]-[LIVM]-x(5,11)-R-[STAQ]-A-x-[LIVMA]-x-[STACV] ("x" means "any amino-acid", "[GAS]" means "any of those inside the brackets" and "x(5,11)" is a gap of length between 5 and 11). The cardinality of this pattern is 10^{22} which is huge but we can see on table 4 that the characteristics of the smallest associated m-unambiguous DFA are far smaller. Of course the number of states grows quickly with m but fortunately, protein sequences are usually modelized with low order Markov chains ($m \leqslant 2$).

We consider now the 1 332 signatures of the PROSITE database (release 19.23) and a dataset consisting of 280 proteins from the SWISS-PROT database Gasteiger et al. (2001) which belongs to the transmembrane type (according to their annotations) with a total length of 84 192 amino-acids. We use the dataset to estimate an independent homogeneous model (order $m = 0$ Markov model) and want to point out significant over-represented PROSITE signatures in our transmembrane sequences.

The 27 signatures which appear at least one time in the transmembrane dataset are listed in table 5. For example, we can see that the signature PS00007 (Tyrosine kinase phosphorylation site) appears 102 times

in the dataset but that the corresponding p-value (0.48) is unsignificant. The signature definition is [RK]-x(2)-[DE]-x(3)-Y or [RK]-x(3)-[DE]-x(2)-Y which gives a cardinality of 25.6 millions, but the number of states (resp. final states) of the smallest unambiguous associated DFA is only $L = 72$ (resp. $F = 19$). The computational time is also given in the table and we can see that it highly depends on the combinatorial complexity of the considered signature ranging from a couple of seconds for the simplest ones to more than one hour for the most complicated one.

In the paper Nicodeme et al. (2002), the author used a DFA approach to compute exact order one and two moments through formal computations and generating functions in the independent case. Using the extension of their method we presented here, we are able to do much more with a dramatic improvement in terms of efficiency.

Two significant signatures are especially interesting because they have a high number of occurrences in the dataset: PS00008 and PS00294. The first one is annotated in the PROSITE database as a N-myristoylation site and the second one as a Prenyl group binding site. It could be interesting to further investigate the biological relevance of this site for transmembrane proteins.

5. Conclusion

In this paper, we push forward the idea of using DFA to produce moment generating functions of pattern random occurrences to the next level. By introducing the formal notion of PMC (proposed along with explicit construction algorithms), we provide an optimal way to perform Markov chain embedding for a wide range of pattern problem.

In order to illustrate the usefulness of the notion of PMC, we explain in detail how we can use it to compute the exact distribution of a pattern using only basic sparse linear algebra and straightforward recurrences. We also compare the numerical complexity of this approach to those of various classical asymptotic approximations (Gaussian, binomial, Poisson and large deviation) for which the PMC framework bring both effectiveness and simplicity.

We finally consider practical applications of these results by considering two examples of highly degenerated pattern problem. The first one concerns structured motifs which distributions have already been studied by Robin et al. (2002); Stefanov et al. (2006).

Despite the fact that our general approach does not consider the problem from the competing patterns point of view (like the previous approaches do), it is nevertheless able to perform the computation up to 100

ID	L	F	N_{obs}	$\mathbb{P}(N \geqslant N_{\text{obs}})$	time (s)
PS01243	1656	10	2	6.6×10^{-14}	48.4
PS01270	270	2	1	5.8×10^{-11}	3.4
PS00556	50	1	2	7.5×10^{-11}	1.4
PS01114	12	1	2	9.5×10^{-11}	0.4
PS01188	14	1	2	1.3×10^{-9}	0.3
PS01218	261	2	2	2.5×10^{-8}	6.4
PS01133	8840	136	1	3.0×10^{-8}	168.0
PS01214	11	1	1	3.4×10^{-6}	0.2
PS01246	1332	40	1	3.4×10^{-6}	20.3
PS00008	64	32	1141	4.9×10^{-6}	1961.6
PS00294	9	3	387	3.2×10^{-5}	56.5
PS01221	427	14	1	1.5×10^{-4}	4.9
PS00004	7	2	129	1.8×10^{-4}	12.3
PS01128	2587	63	1	9.0×10^{-4}	44.9
PS01309	59	2	1	1.1×10^{-3}	0.7
PS00006	12	4	1034	8.1×10^{-3}	406.5
PS00016	4	1	16	2.9×10^{-2}	1.1
PS00009	5	1	53	5.7×10^{-2}	4.6
PS00217	1152	40	1	6.7×10^{-2}	14.2
PS00133	40	3	1	1.1×10^{-1}	0.6
PS00007	72	19	102	1.4×10^{-1}	104.6
PS00001	9	3	398	3.6×10^{-1}	58.8
PS00029	20480	4096	15	4.8×10^{-1}	5173.3
PS00430	17	2	1	7.4×10^{-1}	0.2
PS00017	60	4	2	9.2×10^{-1}	1.5
PS00005	6	2	955	9.4×10^{-1}	240.2
PS00342	5	2	1073	1.0×10^{-0}	548.8

TABLE 5: The 27 PROSITE signatures (out of $1,332$) that appear at least once in the transmembrane dataset. These signatures are ordered by increasing exact p-values computed in reference with an order $m = 0$ Markov model which parameters are estimated on the dataset. L (resp. F) is the number of states (resp. of final states) of the smallest DFA that recognize the pattern. N_{obs} is the number of observed occurrence in the transmembrane dataset and $\mathbb{P}(N \geqslant N_{\text{obs}})$ is the p-value of the observation. The indicated time is the overall running time to build the DFA, count the occurrences and perform the exact p-value computation using a Intel 2.6 GHz P4 workstation. A significance threshold of 3.8×10^{-5} (5% threshold with Bonferroni correction) is represented by a solid line.

times faster than the previous (but not optimized) ones. It is however clear that this approach will not be able to deal with longer gaps without a significant additional computational effort. The counterpart of this drawback is a more flexible method allowing for example to take into account several occurrences in the same sequence or to consider heterogeneous models.

Like in Nicodeme et al. (2002) we also considered the signature from the PROSITE database. As these signature are often built from poorly conserved protein sequences, many of them present high combinatorial complexity. As a consequence, 12% of the PROSITE patterns considered by Nicodeme et al. (2002) was not tractable, the largest automaton successfully processed having 946 states. In the present study however, our more straightforward Markov chain embedding approach allows us to treat all signatures with our largest automaton having 20 480 states which dramatically outperform the previous method.

Finally, let us add that all these results are already implemented in the Statistic for Patterns package (SPatt, freely available at `http://stat.genopole.cnrs.fr/spatt`).

Acknowledgment

I would like to thank Hugues Richard for kindly allowing me to access the structured motifs dataset he used in his article. I also want to warmly thank both anonymous referees for their constructive comments and suggestions.

References

D. L. Antzoulakos. Waiting times for patterns in a sequence of multistate trials. *J. Appl. Prob.*, 38:508–518, 2001.

J. D. Biggins and C. Cannings. Markov renewal processes, counters and repeated sequences in Markov chains. *Adv. Appl. Prob.*, 19:521–545, 1987.

S. Chadjiconstantinidis, D. L. Antzoulakos, and M. V. Koutras. Joint distribution of successes, failures and patterns in enumeration problems. *Adv. Appl. Prob.*, 32:866–884, 2000.

O. Chryssaphinou and S. Papastavridis. The occurrence of a sequence of patterns in repeated dependent experiments. *Theory Prob. Appl.*, 35:167–173, 1990.

T. H. Cormen, C. E. Leiserson, and R. L. Rivest. *Introduction to Algorithms*, chapter 34, pages 853–885. MIT Press, 1990.

M. Crochemore and C. Hancart. *Handbook of Formal Languages, Volume 2, Linear Modeling: Background and Application*, chapter Automata for Matching Patterns, pages 399–462. Springer-Verlag, Berlin, 1997.

M. Crochemore and V. T. Stefanov. Waiting time and complexity for matching patterns with automata. *Info. Proc. Letters*, 87:119–125, 2003.

J. C. Fu. Distribution theory of runs and patterns associated with a sequence of multi-state trials. *Statistica Sinica*, 6(4):957–974, 1996.

J. C. Fu and Y. M. Chang. On probability generating functions for waiting time distributions of compound patterns in a sequence of multistate trials. *J. Appll. Prob.*, 30:183–208, 2002.

J. C. Fu and M. V. Koutras. Distribution theory of runs: a Markov chain approach. *J. Amer. Statist. Assoc*, 89:1050–1058, 1994.

J. C. Fu and W. Y. W. Lou. *Distribution Theory of Runs and Patterns and Its Applications: A Finite Markov Chain Approach*. World Scientific, Singapore, 2003.

E. Gasteiger, E. Jung, and A. Bairoch. SWISS-PROT: Connecting biological knowledge via a protein database. *Curr. Issues Mol. Biol.*, 3:47–55, 2001.

J. Glaz, M. Kulldorff, V. Pozdnyakov, and J. M. Steele. Gambling teams and waiting times for patterns in two-state Markov chains. *J. Appl. Probab.*, 43(1):127–140, 2006.

L. J. Guibas and A. M. Odlyzko. String overlaps, pattern matching and transitive games. *J. Combin. Theory A*, 30:183–208, 1981.

J. E. Hopcroft, R. Motwani, and J. D. Ullman. *Introduction the automata theory, languages, and computation, 2e d.* ACM Press, New York, 2001.

N. Hulo, A. Bairoch, V. Bulliard, L. Cerutti, E. De Castro, P. S. Langendijk-Genevaux, M. Pagni, and C. J. A. Sigrist. The PROSITE database. *Nucleic Acid Rs.*, 34:D227–D230, 2006.

R. B. Lehoucq, D. C. Sorensen, and C. Yang. *ARPACK Users' Guide: Solution of Large-Scale Eigenvalue Problems with Implicitly Restarted Arnoldi Methods.* SIAM, 1998.

S.-Y. R. Li. A martingale approach to the study of occurrence of sequence patterns in repeated experiments. *Ann. Prob.*, 8:1171–1176, 1980.

W. Y. W. Lou. On runs and longest run tests: A method of Finite Markov chain imbedding. *J. Am. Statist. Assoc.*, 91(436):1595–1601, 1996.

L. Marsan and M.-F. Sagot. Algorithms for extracting structured motifs using a suffix tree with an application to promoter consensus identification. *J. Comp. Biol.*, 7:345–362, 2000.

P. Nicodeme, B. Salvy, and P. Flajolet. Motifs statistics. *Theor. Comput. Sci.*, 28(2):593–617, 2002.

G. Nuel. Effective p-value computations using Finite Markov Chain Imbedding (FMCI): application to local score and to pattern statistics. *Algo. Mol. Biol.*, 1(5), 2006a.

G. Nuel. Cumulative distribution function of a geometric Poisson distribution. *J. Stat. Comp. and Sim.*, 2006b. In press, preprint available at http://stat.genopole.cnrs.fr/˜gnuel.

G. Nuel. Numerical Solutions for Patterns Statistics on Markov Chains. *Stat. App. Gen. Mol. Biol.*, 1(17), 2006c.

G. Reinert, S. Schbath, and M. Waterman. Probabilistic ans statistical properties of words, an overview. *J. Comp. Biology*, 7:1–46, 2000.

S. Robin and J.-J. Daudin. Exact distribution of the distances between any occurrences of a set of words. *Ann. INst. Stat. Mat.*, 36:895–905, 2001.

S. Robin and J.-J. Daudin. Exact distribution of word occurrences in a random sequence of letters. *J. Applied Probabilities*, 36:179–193, 1999.

S. Robin, J.-J. Daudin, H. Richard, M.-F. Sagot, and S. Schbath. Occurrence probability of structured motifs in random sequences. *J. Comp. Biol.*, 9:761–773, 2002.

V. T. Stefanov. On some waiting time problems. *J. Appl. Prob.*, 37:756–764, 2000.

V. T. Stefanov. The intersite distances between pattern occurrences in strings generated by general discrete- and continuous-time models: an algorithmic approach. *J. Appl. Prob.*, 40:881–892, 2003.

V. T. Stefanov and A. G. Pakes. Explicit distributional results in pattern formation. *Ann. Appl. Prob.*, 7: 666–678, 1997.

V. T. Stefanov and A. G. Pakes. Explicit distributional results in pattern formation II. *Autral. and New Zealand J. Statist.*, 41:79–90, 1999.

V. T. Stefanov, S. Robin, and S. Schbath. Waiting times for clumps of patterns and for structured motifs in random sequences. *Discrete Applied Mathematics*, 2006.