J. Appl. Prob. **45,** 226–243 (2008)
Printed in England

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

GRÉGORY NUEL,* *University of Evry*

Abstract

In the framework of patterns in random texts, the Markov chain embedding techniques consist of turning 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 using the new notion of pattern Markov chains (PMCs), and we give explicit constructive algorithms to build the PMC associated to any given pattern problem. The interest of PMCs is then illustrated through the exact computation of P-values whose 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 signatures), which further illustrate the usefulness of our approach.

Keywords: Language; regular expression; exact distribution; structured motif; PROSITE signature

2000 Mathematics Subject Classification: Primary 65C40

1. Introduction

The theory concerning pattern and motif occurrence in random strings has been of interest since the 1950s. Computational molecular biology has been a major area of application for this theory since the 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 by Guibas and Odlyzko (1981), Chrysaphinou and Papastavridis (1990), Robin and Daudin (1999), (2001), and Stefanov (2003); Markov chain embeddings have been used by Fu (1996), Chadjiconstantinidis *et al.* (2000), Antzoulakos (2001), and Fu and Chang (2002); Markov renewal embeddings have been used by Biggins and Cannings (1987); exponential families with either Markov chain or Markov renewal embeddings have been used by Stefanov and Pakes (1997), (1999), and Stefanov (2000); and martingale techniques have been used by Li (1980) and 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 noted that the computational effort is substantial for all of the available methods when the pattern cardinality (i.e. the number of strings the pattern contains) becomes relatively large. Taking inspiration from pattern matching theory, Nicodeme *et al.* (2002) first proposed overcoming this problem using deterministic finite automata (DFAs) in order to obtain a moment generating function of pattern counts through the Chomsky

Received 19 June 2006; revision received 28 November 2007.
* Current address: MAP5, UMR CNRS 8145, Université Paris Descartes, 45 rue des Saints-Pères, F-75006 Paris, France. Email address: gregory.nuel@math-info.univ-paris5.fr

and Schützenberger algorithm. A similar approach using exponential families has also been proposed by Crochemore and Stefanov (2003).

The purpose of this paper is to push forward the connection between patterns and automata by introducing an optimal Markov chain embedding through the notion of pattern Markov chains (PMCs) (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 pattern applications where our method proves its practical usefulness (Section 4).

2. PMCs

2.1. Automata and languages

In this section we first introduce some classical definitions and results from the well-known theory of languages and automata; see Hopcroft *et al.* (2001).

We consider a *finite alphabet* $\mathcal{A} = \{a_1, \ldots, a_k\}$ whose 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 $\mathcal{A} = \{\text{a, b}\}$ and $\mathcal{L} = \{\text{ab, abbaba, bbbbb}\}$ is a language over $\mathcal{A}$.

The *product* $\mathcal{L}_1\mathcal{L}_2$ of two languages is the language $\{w_1w_2, w_1 \in \mathcal{L}_1, w_2 \in \mathcal{L}_2\}$, where w_1w_2 is the concatenation (or product) of w_1 and w_2. If $\mathcal{L}$ is a language, $\mathcal{L}^n = \{w_1 \cdots w_n$ with $w_1, \ldots, w_n \in \mathcal{L}\}$, and the *star closure* of $\mathcal{L}$ is defined by $\mathcal{L}^* = \bigcup_{n \geqslant 0} \mathcal{L}^n$. Hence, the language $\mathcal{A}^*$ is the set of all possible words over $\mathcal{A}$. For example, we have $\{\text{ab}\}\{\text{abbaba, bbbbb}\} = \{\text{ababbaba, abbbbbb}\}$, $\{\text{ab}\}^3 = \{\text{ababab}\}$, and $\{\text{ab}\}^* = \{\varepsilon, \text{ab, abab, } \ldots\}$.

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

Definition 1. If $\mathcal{A}$ is a finite alphabet, $\mathcal{Q}$ is a finite set of states, $s \in \mathcal{Q}$ is a starting state, $\mathcal{F} \subset \mathcal{Q}$ is a subset of the final states, and $\delta : \mathcal{Q} \times \mathcal{A} \to \mathcal{Q}$ is a transition function, then $(\mathcal{A}, \mathcal{Q}, s, \mathcal{F}, \delta)$ is a *deterministic finite automaton* (DFA). For all $a = a_1 \cdots a_{d-1}a_d \in \mathcal{A}^d$, $d \geqslant 2$, and $q \in \mathcal{Q}$, we recursively define $\delta(q, a_1 \cdots a_{d-1}a_d) = \delta(\delta(q, a_1 \cdots a_{d-1}), a_d)$. A word $w \in \mathcal{A}^h$ 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 Figure 1 for a graphical representation of a DFA.

We can now give the most important result of this subsection, which is a simple application of the classical Kleene theorem and Rabin and Scott theorem; see Hopcroft *et al.* (2001).

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

2.2. Connection with patterns

We define the *pattern* over the finite alphabet $\mathcal{A}$ to be any finite language over the same alphabet such that no element is included in another element (this last condition is used to simplify many definitions and results in avoiding degenerate 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 $\mathcal{A} = \{\text{a, b}\}$ then the smallest DFA associated with the pattern $\mathcal{W}_1 = \text{ab}\mathcal{A}^1\text{aa}\mathcal{A}^1\text{ab}$ has $L = 12$ states and $F = 1$ final state. A graphical representation of this DFA is given in Figure 1.

https://doi.org/10.1239/jap/1208358964 Published online by Cambridge University Press

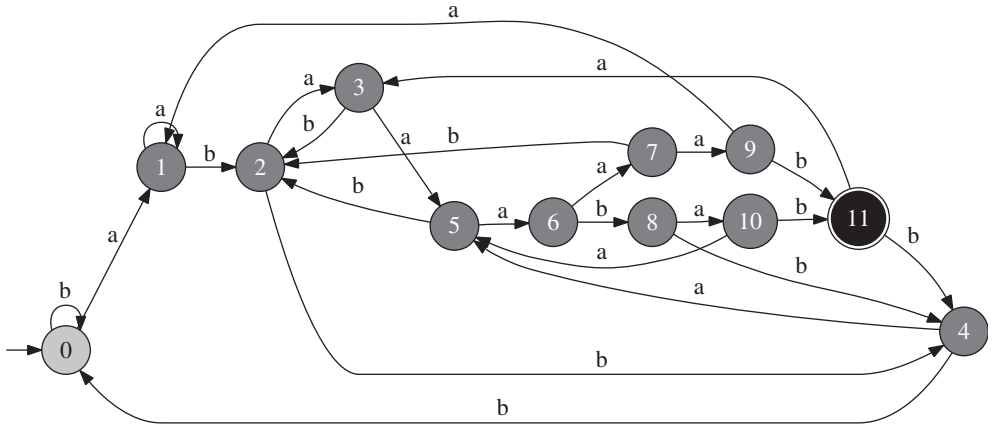

FIGURE 1: Graphical representation of the DFA $(\mathcal{A}, \mathcal{Q}, s, \mathcal{F}, \delta)$ with $\mathcal{A} = \{\mathrm{a}, \mathrm{b}\}$, $\mathcal{Q} = \{0, 1, 2, \ldots, 10, 11\}$, $s = 0$, $\mathcal{F} = \{11\}$, and $\delta(0, \mathrm{a}) = 1$, $\delta(0, \mathrm{b}) = 0$, $\delta(1, \mathrm{a})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

It is well known from 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.

We should note 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 $\mathcal{W} = \{w = w_1 \cdots w_h\}$ is a single word of length h then its smallest associated DFA is of size $L = h + 1$ and defined by $\mathcal{Q} = \{\varepsilon, w_1, w_1w_2, \ldots, w\}$, 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 the case where $\mathcal{Q}$ is the union of all pattern prefixes), but it would not necessarily be the smallest one. In order to be more efficient in the DFA design, we should instead use the classical and well-known algorithms provided by the theory of 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 cardinality of the pattern. The pattern $\mathcal{W}_1$ is recognized by the DFA of Figure 1, $\mathcal{W}_2$ is recognized by the DFA of Figure 2, and $\mathcal{W}_{11}$, a pattern with a cardinality of several millions, is recognized by a DFA having only a few thousand states.

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

Theorem 2. *If $X = X_1X_2 \cdots X_i \cdots$ is an independent and identically distributed (i.i.d.) sequence on $\mathcal{A}$, if $\mathcal{W}$ is a pattern, and if $(\mathcal{A}, \mathcal{Q}, s, \mathcal{F}, \delta)$ is an associated DFA, then the sequence*

https://doi.org/10.1239/jap/1208358964 Published online by Cambridge University Press

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 is the number of final states.

k	$\lvert\mathcal{W}_k\rvert$	L	F
1	4	12	1
2	16	27	3
3	64	57	6
4	256	122	13
5	1 024	262	28
6	4 096	562	60
7	16 384	1 207	129
8	65 536	2 592	277
9	262 144	5 567	595
10	1048 576	11 957	1278
11	4194 304	25 682	2745

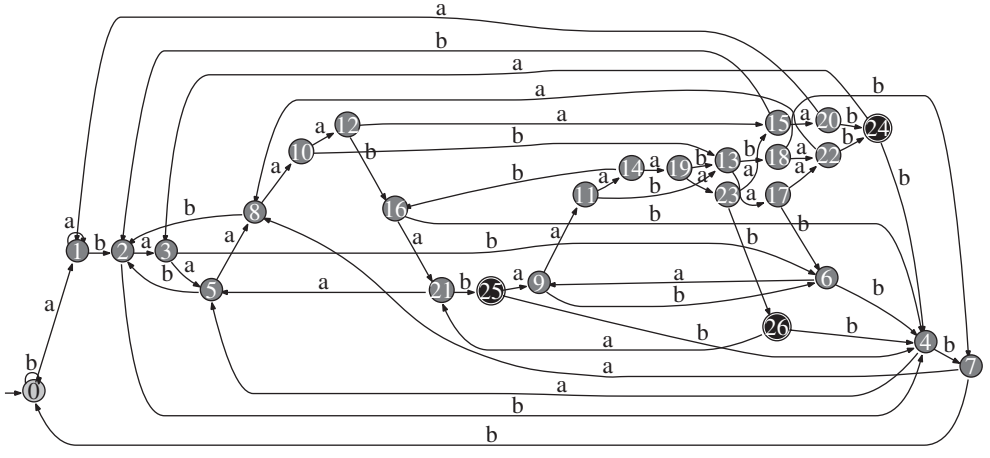

FIGURE 2: Graphical representation of the smallest DFA associated with $\mathcal{W}_2 = \mathrm{ab}\mathcal{A}^2\mathrm{aa}\mathcal{A}^2\mathrm{ab}$. This DFA has $L = 27$ states and $F = 3$ final states.

$Y = Y_0Y_1Y_2\cdots Y_i$ defined by

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

*is an order-*1 *Markov chain with transition matrix*

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

and such that 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 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 $X_1 \cdots X_i$ ends with an occurrence of

https://doi.org/10.1239/jap/1208358964 Published online by Cambridge University Press

the pattern and is therefore an element of $\mathcal{A}^*\mathcal{W}$, and is thus 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 there exists a set $\mathcal{Q}$, a subset $\mathcal{F} \subset \mathcal{Q}$, and a function $G\colon \mathcal{A}^* \to \mathcal{Q}^*$ such that

(i) for all $x \in \mathcal{A}^*$, we let $y = G(x)$; for all $0 \leqslant i \leqslant |x|$, $\mathcal{W}$ ends at position i in x, which is equivalent to $y_i \in \mathcal{F}$;

(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 recursively define the function $\tilde{G}\colon \mathcal{A}^* \to \mathcal{Q}^*$ by $\tilde{G}(\varepsilon) = G(\varepsilon)$ and $\tilde{G}(xa) = \tilde{G}(x)\Delta(x, a)$. We now define $\tilde{\Delta}(\tilde{G}(x), a) = \Delta(x, a)$ on the quotient space $(\mathcal{A}^*)_{\mathcal{R}}$, where $x\mathcal{R}x'$ is equivalent to $\tilde{G}(x) = \tilde{G}(x')$.

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

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

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 in Figure 1. If X is the original sequence, we build the PMC Y as follows (final states are given 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 ending in position 18 (abbaabab, overlapping the previous occurrence). The transition matrix of Y is given by

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

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

As explained in the introduction, Nicodeme *et al.* (2002) proposed using the pattern's DFA to obtain the pattern generating function through the Chomsky and Schützenberger algorithm,

https://doi.org/10.1239/jap/1208358964 Published online by Cambridge University Press

and derived from it exact results and asymptotic moments. More recently, Crochemore and Stefanov (2003) used the pattern's automaton conjointly with exponential family results with the same aim. Instead of focusing on the generating function only (as done in these papers), here we propose a more straightforward and practical approach in which we exploit our new PMC to improve a wide range of classical pattern methods.

2.3. Extensions

The methods we have presented until now are valid only for overlapping occurrences of a pattern in an i.i.d. sequence. Here we propose 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 $(\mathcal{A}, \mathcal{Q}, \mathcal{F}, s, \delta)$ in which there exist $q \in \mathcal{Q}$ and $a, b \in \mathcal{A}^m$ such that $a \neq b$ and $\delta(q, a) = \delta(q, b)$ is called *m-ambiguous*. A DFA which is not m-ambiguous is called *m-unambiguous*.

Please note that the m-ambiguity presented here is different from the classical notion of *ambiguity* for DFAs (meaning that there exist two different paths 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 notation:

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

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

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

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

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

is an order-1 Markov chain with transition matrix

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

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

Proof. The proof is very similar to the i.i.d. case (Theorem 2) except that the m-unambiguity is obviously required to ensure that all the $\delta^{-m}(p)$ are singletons.

Using this theorem, it is possible to apply all preceding methods to Markovian sequences. But the key question is of course, is it possible to build an m-unambiguous pattern DFA and if so, how?

Nicodeme *et al.* (2002, Algorithm 6) showed that this could be achieved starting from a DFA associated with the pattern by duplicating states until all the ambiguities are 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.

https://doi.org/10.1239/jap/1208358964 Published online by Cambridge University Press

As suggested by Nicodeme *et al.* (2002), Algorithm 1 simply duplicates states for which there exists an m-ambiguity while preserving the DFA's ability to recognize its language. As only the necessary states are duplicated, this algorithm also preserves the optimality of the produced DFA.

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

Require: $A = (\mathcal{A}, \mathcal{Q}, s, \mathcal{F}, \delta)$ is an $(m-1)$-unambiguous DFA that recognizes $\mathcal{W}$.
1: Initialization:
2: $\mathcal{Q}_0 = \mathcal{Q}$ for all $q \in \mathcal{Q}$, $\mathcal{D}_q = \delta^{-m}(q)$, and $\mathcal{G}_q = \Delta^{-1}(q)$
3: Main loop:
4: for all $q \in \mathcal{Q}_0$ do
5: while $|\mathcal{D}_q| > 1$ do
6: take $a = a_1 \cdots a_m \in \mathcal{D}_q$
7: add a new state q_a to $\mathcal{Q}$
8: if $q \in \mathcal{F}$ then add q_a to $\mathcal{F}$
9: define $\mathcal{D}_{q_a} = \{a\}$ and $\mathcal{G}_{q_a} = \varnothing$
10: for all $b \in \mathcal{A}$ do $\delta(q_a, b) = \delta(q, b)$ and add q_a to $\mathcal{G}_{\delta(q,b)}$
11: for all $p \in \mathcal{G}_q$
12: if $\delta(p, a_m) = q$ and $\delta^{-(m-1)}(p) = a_1 \cdots a_{m-1}$ (empty condition if $m = 1$) then
13: $\delta(p, a_m) = q_a$ and add p to $\mathcal{G}_{q_a}$
14: for all $p \in \mathcal{G}_q$, if $q \notin \delta(p, \mathcal{A})$ then remove q from $\mathcal{G}_q$
15: remove a from $\mathcal{D}_q$

In order to achieve m-unambiguity we could hence successively remove 1-ambiguity, then 2-ambiguity, and so on until we finally remove m-ambiguity having used a total of m applications of Algorithm 1. For example, we can use this approach to transform the 2-ambiguous DFA of Figure 1 ($\delta^{-2}(1) = \{\text{aa}, \text{ba}\}$) into the 2-unambiguous DFA of Figure 3; in the process, Algorithm 1 replaces state 1 by states 1 and 12.

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

Adapting pattern methods to such kinds 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 to 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*

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

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

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

https://doi.org/10.1239/jap/1208358964 Published online by Cambridge University Press

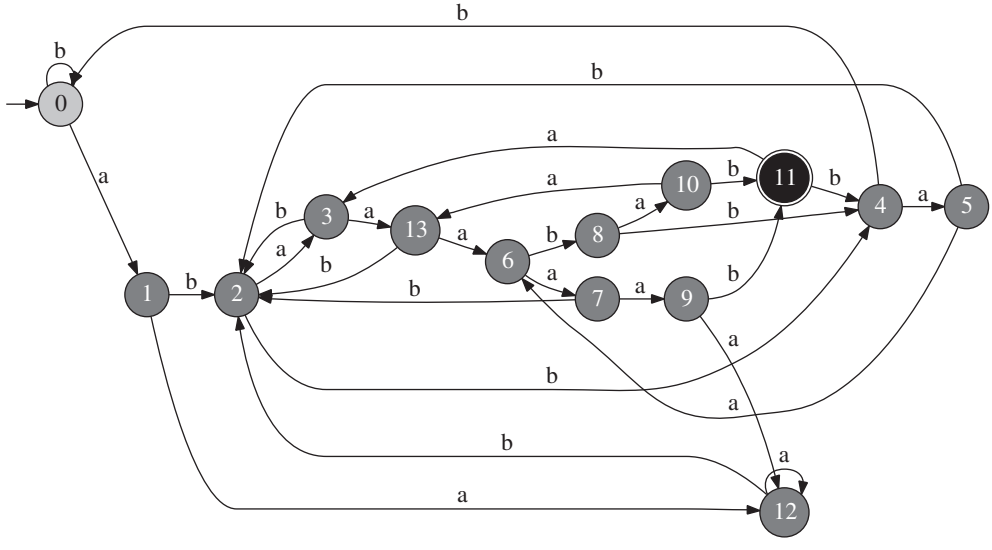

FIGURE 3: Graphical representation of the smallest 2-unambiguous DFA associated with $\mathcal{W}_1 = \mathrm{ab}\mathcal{A}^1\mathrm{aa}\mathcal{A}^1\mathrm{ab}$ and $\mathcal{A} = \{\mathrm{a}, \mathrm{b}\}$. This DFA has been built from the DFA of Figure 1 using Algorithm 1.

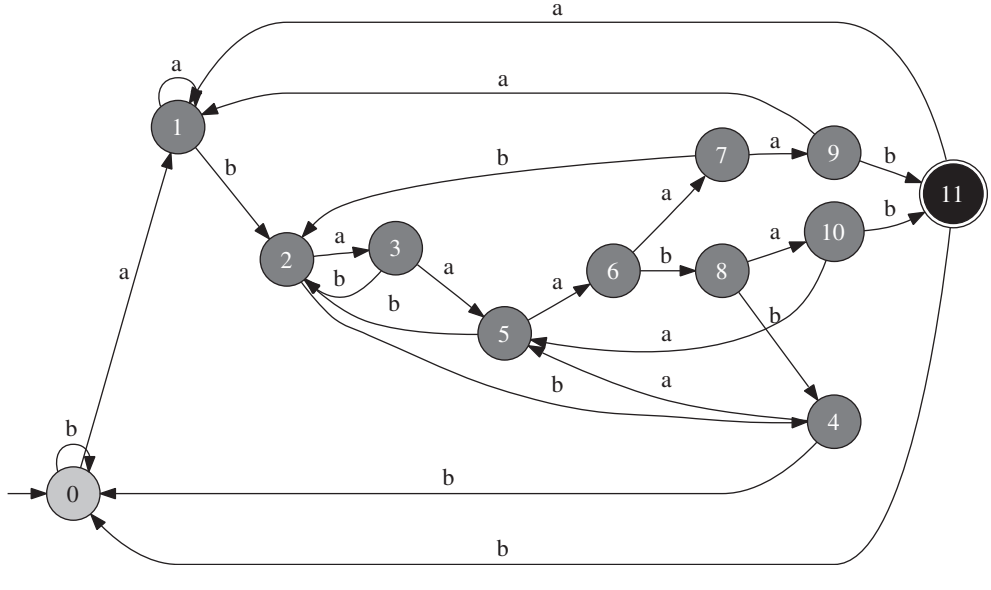

FIGURE 4: Graphical representation of a renewal DFA associated with $\mathcal{W}_1 = \mathrm{ab}\mathcal{A}^1\mathrm{aa}\mathcal{A}^1\mathrm{ab}$ and $\mathcal{A} = \{\mathrm{a}, \mathrm{b}\}$. This DFA has been built from the DFA of Figure 1 using Proposition 2.

An application of Proposition 2 to the DFA of Figure 3 gives the DFA of Figure 4.

Once this transformation has been carried out, all previous results will hold for renewal occurrences using our modified DFA. We should note that when doing so, the pattern self-

https://doi.org/10.1239/jap/1208358964 Published online by Cambridge University Press

overlapping matrix is obviously null and, hence, makes compound Poisson approximations easier to use as they are only simple Poisson approximations.

We can also extend the notion of renewal occurrences to the notion 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 occurrences are exactly the same). In order to consider d-renewal occurrences of a pattern $\mathcal{W}$ we simply need to count the renewal occurrences of $\mathcal{W}\mathcal{A}^d$.

3. Using PMCs

3.1. Exact distribution

DFAs have been used by Nicodeme *et al.* (2002) and Crochemore and Stefanov (2003) to obtain moment generating functions of the number of occurrences of any pattern in a random sequence. With the help of efficient numerical algorithms (e.g. fast Taylor expansions), it is therefore possible to 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 subsection we consider a more direct approach by showing how we can use PMCs efficiently to compute exact P-values. Our approach firstly consists of producing, through a PMC, an optimal Markov chain embedding of the problem and then using recurrence relations which exploit the sparse structure of the transition matrix to perform the computations.

The technique of Markov chain embedding (also called finite Markov chain embedding (FMCI)) has been used for pattern problems by Fu and Koutras (1994), Lou (1996), and Fu and Lou (2003). If many embedded Markov chains can be built for a given problem, the design of a space-efficient one is of course of critical interest for practical applications. Here we propose to solve this problem by showing the very simple connection that exists between PMCs and FMCIs.

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

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 $X_1 \cdots X_j$.

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

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

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

$$\boldsymbol{R}_{i,j} = \begin{cases} \boldsymbol{P} & \text{if } i = j, \\ \boldsymbol{Q} & \text{if } i + 1 = j, \\ \mathbf{0} & \text{otherwise}, \end{cases} \qquad \text{and} \qquad \boldsymbol{v}_i \equiv \mathbf{0} \ \text{ for } 1 \leqslant i < c \ \text{ and } \ \boldsymbol{v}_c = \mathbf{\Sigma}_{\boldsymbol{Q}},$$

where $\mathbf{\Sigma}_{\boldsymbol{Q}}$ is the column vector resulting from the sum of $\boldsymbol{Q}$.

https://doi.org/10.1239/jap/1208358964 Published online by Cambridge University Press

Proof. The proof is straightforward since transitions in $\boldsymbol{P}$ will not increment the number of occurrences while transitions in $\boldsymbol{Q}$ will increment them by one.

Example 2. If $c = 2$, we obtain the following transition matrix:

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

As proposed in Nuel (2006a), it is therefore possible to obtain 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*

$$\mathrm{P}(N_n < c \mid X_1 = i) = (\boldsymbol{u}^{n-1})_i \quad \textit{and} \quad \mathrm{P}(N_n \geqslant c \mid X_1 = i) = \sum_{j=0}^{n-2} (\boldsymbol{v}^j)_i,$$

where $(\,)_i$ denotes the ith component of a vector. For $\boldsymbol{x} = \boldsymbol{u}$ or $\boldsymbol{x} = \boldsymbol{v}$, we have the size-$L$-block decomposition

$$\boldsymbol{x}^j = (x^j_{(c-1)}, \ldots, x^j_0)' \quad \textit{for all } j \geqslant 0,$$

and we have the recurrence relations

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

with $\boldsymbol{u}^0$ a column vector of 1*s and $\boldsymbol{v}^0 = \boldsymbol{v}$.*

3.2. Asymptotic approximations

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

Table 2 summarizes the time and memory complexities for all these approaches. Let us first point out that the alphabet size k and the cardinality L of the PMC state space are critical parameters for all the approaches since $k \times L$, the number of nonzero terms in the transition matrix of the PMC, is the complexity of a sparse product of this matrix with a vector.

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

Method	Memory complexity	Time complexity
Exact	$k \times L + N_{\mathrm{obs}} \times L$	$k \times L \times N_{\mathrm{obs}} \times n$
Gaussian	$k \times L + F \times L$	$k \times L + F \times L \times \log n + F^2$
Binomial/Poisson	$k \times L$	$k \times L + F + \log N_{\mathrm{obs}}$
Geometric Poisson	$k \times L + F^2$	$k \times L + F^2 + N_{\mathrm{obs}}$
Compound Poisson	$k \times L + F^2 + N_{\mathrm{obs}}$	$k \times L + F^2 + N_{\mathrm{obs}}^2$
Large deviations	$k \times L$	$k \times L$

https://doi.org/10.1239/jap/1208358964 Published online by Cambridge University Press

Unlike with the exact approach, we have to assume both homogeneity and ergodicity of the underlying Markov sequence in order to obtain these approximations. It is then possible to compute exact first-order and second-order moments of the pattern count with a constant N_{obs} complexity and only a logarithm n complexity, thus resulting in a computational improvement over the Markov embedding approach. We should, however, note that the number of final states F appears in both the memory and the time complexity in a linear or quadratic form.

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

If we now turn 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 requires a quadratic complexity in 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 in N_{obs} thanks to the recurrence formulae given in Nuel (2007).

Finally, large deviation approximations display the smallest complexities as they only rely on sparse products to solve eigenproblems related to the transition matrix of the PMC (which can be carried out efficiently with the Arnoldi class algorithm; see Lehoucq *et al.* (1998)). However, it is necessary to emphasize that in practice the large deviation approaches are slower than other approximations (but also more reliable for exceptional patterns).

4. Applications

In this section we propose to illustrate the interest of PMCs through two examples of highly degenerated biological patterns.

4.1. Structured motifs

Here we consider 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 of a sequence of two or more strings, each occurrence of which is separated by a specific number of letters. For example, the structured pattern $\mathtt{ttgaca}\mathcal{A}^{16:18}\mathtt{tataata}$ is composed of two strings separated by at least 16 and at most 18 letters. Robin *et al.* (2002) first gave a Poisson approximation to the problem, and more recently, Stefanov *et al.* (2007) proposed exact methods to compute the exact distribution of this kind of pattern. 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 sequences of length 100 located in the upstream region of 131 genes of the bacterium *Bacillus 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 and w_2 are two strings and d_1 and d_2, $d_1 \leqslant d_2$, are two integers.

For technical considerations, Stefanov *et al.* (2007) excluded occurrences of the structured motif where w_1 or 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 motifs) are obviously closely related.

https://doi.org/10.1239/jap/1208358964 Published online by Cambridge University Press

Assuming that the 131 sequences (note that in Stefanov *et al.* (2007) a typographical error indicated that the dataset contained 130 sequences and not the 131 sequences which they used in all their subsequent binomial computations) are generated according to a homogeneous Markov model whose parameters are estimated from the dataset, we consider the random variables $(N_i)_{1\leqslant i\leqslant 131}$ and N_i', counts of the numbers of occurrences of the pattern and restricted pattern defined above in the ith sequence, respectively. Hence, we consider $N = \sum_{i=1}^{131} N_i$ and $M = \sum_{i=1}^{131} \mathbf{1}_{N_i \geqslant 1}$ (as well as their restricted versions, N' and M').

Table 3 lists the 15 most significant structured motifs among the 71 that have been tested. The $\mathrm{P}_s(M' \geqslant \text{obs})$ column is the same as the last column of Table 5 of Stefanov *et al.* (2007) except for two structured motifs whose numbers of occurrences have somehow been miscounted by the authors (ttgaca$\mathcal{A}^{16:18}$atataat and gttgaca$\mathcal{A}^{16:18}$tataata, which appear in the sequences rpmH, TrnS, and veG, and rpmH and f82129, respectively, were only observed twice and once, respectively, by Stefanov *et al.* (2007)).

As M and M' are different countings, it is not surprising 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 than 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 difference 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 appears in 88 sequences of the dataset, but its total number of occurrences is 111; the corresponding P-values are

$$\mathrm{P}(M \geqslant 88) = 1.66 \times 10^{-2} \quad \text{and} \quad \mathrm{P}(N \geqslant 111) = 3.50 \times 10^{-4}.$$

Even if the cardinality of each of these structured motifs,

$$|\mathcal{W}| = 4^{16} + 4^{17} + 4^{18} = 90\,194\,313\,216 \simeq 9 \times 10^{10},$$

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

$\mathcal{W}$	$L(F)$	obs	$\mathrm{P}_s(M' \geqslant \text{obs})$	$\mathrm{P}_s(M \geqslant \text{obs})$	$\mathrm{P}(M \geqslant \text{obs})$	$\mathrm{P}(N \geqslant \text{obs})$
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	–	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	–	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}

https://doi.org/10.1239/jap/1208358964 Published online by Cambridge University Press

is huge, we can see that the size of the smallest associated DFA is far smaller, with an order of magnitude of 1000. This of course allows our PMC approach to be very efficient both in terms of memory usage and running time. For example, computing the 71 P-values of the type $\mathrm{P}_s(M \geqslant \text{obs})$ requires a total of 25 seconds on an Intel 2.6 GHz P4® workstation, while the computations of $\mathrm{P}_s(M' \geqslant \text{obs})$ with the previous method requires 3277 seconds on an IBM F80 computer. Therefore, our approach is 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.* (2007) were not designed for numerical performance. Moreover, Stefanov *et al.* (2007) considered the problem to be that of two competing patterns rather than a single (highly degenerated) problem, which results in a marginal increase of complexity with the gap length, while the single-pattern approach presented here is geometrically dependent on this parameter.

We should note that it is 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 $\mathcal{F}_1$ and $\mathcal{F}_2$ contain the final states associated with the patterns w_1 and w_2, respectively. Then, if we consider the corresponding decomposition of the transition matrix

$$\mathbf{\Pi} = \boldsymbol{P} + \boldsymbol{Q}_1 + \boldsymbol{Q}_2,$$

it is possible to obtain the distribution of a structured pattern in a very straightforward way:

$$\mathrm{P}(w_1 \mathcal{A}^d w_2 \text{ starts in } i) = \underbrace{\mu_m \boldsymbol{P}^{i-m}}_{\text{up to } i} \times \underbrace{\boldsymbol{P}^{|w_1|-2} \boldsymbol{Q}_1}_{w_1} \times \underbrace{\boldsymbol{P}^{d}}_{\text{gap}} \times \underbrace{\boldsymbol{P}^{|w_2|-1} \boldsymbol{Q}_2}_{w_2} \times \mathrm{e}_{\mathcal{F}_2}^T.$$

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 us to count both w_1 and w_2 has $L = 16$ states (while the DFA associated with the full structured motif has $L = 1527$ states), and we obtain

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

which is very close to the exact solution (3.02×10^{-5} in Stefanov *et al.* (2007)) despite the fact that important dependencies are not taken into account here.

This alternative approach obviously needs more work in order to deal rigorously with the problem, but it already seems appealing since it combines the interests of both the existing method and the new method. Indeed, most of the complex combinatorial aspects of the problem are embedded in the PMC (whose state space is greatly reduced) and, as in Stefanov *et al.* (2007), dealing with larger gaps is not a problem.

Finally, let us add that our PMC approach to structured motifs has several natural extensions which are likely to be difficult to obtain 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.

https://doi.org/10.1239/jap/1208358964 Published online by Cambridge University Press

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

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

and the heterogeneous (and arbitrary) transition matrix is given by

$$\begin{aligned}\boldsymbol{\pi}_i(a,b) &= \mathrm{P}(X_i = b \mid X_{i-1} = a)\\ &= \frac{(100-i)}{98}\boldsymbol{\pi}^0(a,b) + \frac{(i-2)}{98}\boldsymbol{\pi}^1(a,b) \quad \text{for all } a, b \in \mathcal{A} \text{ and for all } 2 \leqslant i \leqslant 100,\end{aligned}$$

where

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

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 a homogeneous model or according to the heterogeneous model defined above:

$$\begin{aligned}&\mathrm{P}(N(\mathtt{ttgactt}\mathcal{A}^{16:18}\mathtt{ataataa}) \geqslant 1)\\ &\quad = \begin{cases} 6.863712 \times 10^{-6} & \text{with the homogeneous transitions } \boldsymbol{\pi}^0,\\ 8.795492 \times 10^{-8} & \text{with the homogeneous transitions } \boldsymbol{\pi}^1,\\ 1.549870 \times 10^{-6} & \text{with the heterogeneous transitions.}\end{cases}\end{aligned}$$

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 functional families. As proteins 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 (entry PS00889 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 in 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 modeled with low-order Markov chains ($m \leqslant 2$).

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

m	L	F
0	329	30
1	1 393	78
2	10 688	633
3	134 746	3045

https://doi.org/10.1239/jap/1208358964 Published online by Cambridge University Press

TABLE 5: The 27 PROSITE signatures (out of 1332) that appear at least once in the transmembrane dataset. These signatures are ordered by increasing exact P-values computed in reference to an order-($m = 0$) Markov model whose parameters are estimated from the dataset. Here L and F are the numbers of states and final states, respectively, of the smallest DFA that recognizes the pattern, N_{obs} is the number of observed occurrences in the transmembrane dataset, and $\mathrm{P}(N \geqslant N_{\text{obs}})$ 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 an 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.

Signature	L	F	N_{obs}	$\mathrm{P}(N \geqslant N_{\text{obs}})$	Time (s)
PS01243	1 656	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	20 480	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

Now we consider the 1332 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 overrepresented PROSITE signatures in our transmembrane sequences.

The 27 signatures that appear at least once in the transmembrane dataset are listed in Table 5. For example, we can see that the signature PS00007 (Tyrosine kinase phosphorylation site)

https://doi.org/10.1239/jap/1208358964 Published online by Cambridge University Press

appears 102 times in the dataset but that the corresponding P-value (0.14) is insignificant. 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 million, but the numbers of states and final states of the smallest unambiguous associated DFA are only $L = 72$ and $F = 19$, respectively. The computational time is also given in Table 5, 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.

Nicodeme *et al.* (2002) used a DFA approach to compute exact order-1 moments and order-2 moments through formal computations and generating functions in the independent case. Using the extension of their method which we have presented here, we are able to do much more with a dramatic improvement in terms of efficiency.

Signatures PS00008 and PS00294 are especially interesting because they have a high number of occurrences in the dataset. The first one is annotated in the PROSITE database as an 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 pushed the idea of using DFAs to produce moment generating functions of random pattern = occurrences to the next level. By introducing the formal notion of a PMC (proposed along with explicit construction algorithms), we provided an optimal way to perform Markov chain embedding for a wide range of pattern problems.

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

We finally considered practical applications of these results by considering two examples of highly degenerated pattern problems. The first one concerned structured motifs whose distributions have already been studied by Robin *et al.* (2002) and Stefanov *et al.* (2007).

Despite the fact that our general approach did not consider the problem from the competing-patterns point of view (like the previous approaches did), it was nevertheless able to perform the computation up to 100 times faster than the previous (unoptimized) ones. However, it is clear that this approach will not be able to deal with longer gaps without significant additional computational effort. The counterpart of this drawback is a more flexible method allowing, for example, one to take into account several occurrences in the same sequence or to consider heterogeneous models.

As in Nicodeme *et al.* (2002), we also considered signatures from the PROSITE database. As these signatures 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) were not tractable, the largest automaton successfully processed having 946 states. In the present study, however, our more straightforward Markov chain embedding approach allowed us to treat all signatures, with our largest automaton having 20 480 states, which dramatically outperformed 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).

https://doi.org/10.1239/jap/1208358964 Published online by Cambridge University Press

Acknowledgements

I would like to thank Hugues Richard for kindly allowing me to access the structured motifs dataset he used in Robin *et al.* (2002). I also want to warmly thank both anonymous referees for their constructive comments and suggestions.

References

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

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

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

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

Crochemore, M. and Hancart, C. (1997). Automata for matching patterns. In *Handbook of Formal Languages*, Vol. 2, *Linear Modeling: Background and Application*, Springer, Berlin, pp. 399–462.

Crochemore, M. and Stefanov, V. T. (2003). Waiting time and complexity for matching patterns with automata. *Inf. Proc. Lett.* **87,** 119–125.

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

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

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

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

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

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

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

Hopcroft, J. E., Motwani, R. and Ullman, J. D. (2001). *Introduction to Automata Theory, Languages, and Computation*, 2nd edn. ACM Press, New York.

Hulo, N. *et al.* (2006). The PROSITE database. *Nucleic Acid Res.* **34,** D227–D230.

Lehoucq, R. B., Sorensen, D. C. and Yang, C. (1998). *ARPACK Users' Guide. Solution of Large-Scale Eigenvalue Problems with Implicitly Restarted Arnoldi Methods*. Society for Industrial and Applied Mathematics, Philadelphia, PA.

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

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

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

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

Nuel, G. (2006a). Effective p-value computations using finite Markov chain imbedding (FMCI): application to local score and to pattern statistics. *Algo. Molec. Biol.* **1**.

Nuel, G. (2006b). Numerical solutions for patterns statistics on Markov chains. *Statist. Appl. Gen. Molec. Biol.* **5.**

Nuel, G. (2007). Cumulative distribution function of a geometric Poisson distribution. *J. Statist. Comput. Simul.* **77**.

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

Robin, S. and Daudin, J.-J. (1999). Exact distribution of word occurrences in a random sequence of letters. *J. Appl. Prob.* **36,** 179–193.

Robin, S. and Daudin, J.-J. (2001). Exact distribution of the distances between any occurrences of a set of words. *Ann. Inst. Statist. Math.* **36**, 895–905.

Robin, S. *et al.* (2002). Occurrence probability of structured motifs in random sequences. *J. Comput. Biol.* **9,** 761–773.

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

STEFANOV, V. T. (2003). 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.

STEFANOV, V. T. AND PAKES, A. G. (1997). Explicit distributional results in pattern formation. *Ann. Appl. Prob.* **7,** 666–678.

STEFANOV, V. T. AND PAKES, A. G. (1999). Explicit distributional results in pattern formation. II. *Austral. N. Z. J. Statist.* **41,** 79–90, 254.

STEFANOV, V. T., ROBIN, S. AND SCHBATH, S. (2007). Waiting times for clumps of patterns and for structured motifs in random sequences. *Discrete Appl. Math.* **155,** 868–880.