Counting Patterns in Degenerated Sequences

Grégory Nuel

MAP5, CNRS 8145, University Paris Descartes,
45 rue des Saint-Pères, F-75006 Paris, France
gregory.nuel@parisdescartes.fr

Abstract. Biological sequences like DNA or proteins, are always obtained through a sequencing process which might produce some uncertainty. As a result, such sequences are usually written in a degenerated alphabet where some symbols may correspond to several possible letters (ex: IUPAC DNA alphabet). When counting patterns in such degenerated sequences, the question that naturally arises is: how to deal with degenerated positions ? Since most (usually 99%) of the positions are not degenerated, it is considered harmless to discard the degenerated positions in order to get an observation, but the exact consequences of such a practice are unclear. In this paper, we introduce a rigorous method to take into account the uncertainty of sequencing for biological sequences (DNA, Proteins). We first introduce a Forward-Backward approach to compute the marginal distribution of the constrained sequence and use it both to perform a Expectation-Maximization estimation of parameters, as well as deriving a heterogeneous Markov distribution for the constrained sequence. This distribution is hence used along with known DFA-based pattern approaches to obtain the exact distribution of the pattern count under the constraints. As an illustration, we consider a EST dataset from the EMBL database. Despite the fact that only 1% of the positions in this dataset are degenerated, we show that not taking into account these positions might lead to erroneous observations, further proving the interest of our approach.

Keywords: Forward-Backward algorithm, Expectation-Maximization algorithmn, Markov chain embedding, Deterministic Finite state Automaton.

1 Introduction

Biological sequences like DNA or proteins, are always obtained through a sequencing process which might produce some uncertainty. As a result, such sequences are usually written in a degenerated alphabet where some symbols may correspond to several possible letters. For example, the IUPAC [1] protein alphabet includes the following degenerated symbols: `X` for "any amino-acid", `Z` for "glutamic acid or glutamine", and `B` for "Aspartic acid or Asparagine". For DNA sequences, there is even more of such degenerated symbols which exhaustive list and meaning are given in Table 1 along with observed frequencies in several datasets from the EMBL [2] database.

V. Kadirkamanathan et al. (Eds.): PRIB 2009, LNBI 5780, pp. 222–232, 2009.

Table 1. Meaning and frequency of the IUPAC [1] DNA symbols in several files of the release 97 of the EMBL nucleotide sequence database [2]. Degenerated symbols (lowest part of the table) contribute to 0.5% to 1% of the data.

symbol	meaning	est_pro_01	htg_pro_01	htc_fun_01	std_hum_21
A	Adenine	67459	1268408	1347782	1190205
C	Cytosine	53294	1706478	1444861	1031369
G	Guanine	54194	1719016	1325070	809651
T	Thymine	66139	1277939	1334061	1067933
U	Uracil	0	0	0	0
R	Purine (A or G)	13	0	7	39
Y	Pyrimidine (C, T, or U)	6	0	9	37
M	C or A	2	0	6	31
K	T, U, or G	6	0	5	30
W	T, U, or A	6	0	8	26
S	C or G	21	0	4	28
B	not A	0	0	0	0
D	not C	3	0	0	0
H	not G	0	0	1	0
V	not G, not U	0	0	0	0
N	any base	1792	115485	28165	19272

When counting patterns in such degenerated sequences, the question that naturally arise is: how to deal with degenerated positions ? Since most (usually 99%) of the positions are not degenerated, it is usually considered harmless to discard the degenerated positions in order to get an observation. Another option might be to preprocess the dataset by replacing each special letter by the most likely compatible symbol at the position (in reference with some background model). Finally, one might come up with some *ad hoc* counting rule like: "whenever the pattern might occurs I add one[1] to the observed count". However practical, all these solutions remain quite unsatisfactory from the statistician point of view and their possible consequences (like adding or missing occurrences) remain unclear.

In this paper, we want to deal rigorously with the problem of degenerated symbols in sequences by introducing the distribution of sequences under the uncertainty of their sequencing, and then by using this distribution to study the "observed" number of occurrences of a pattern of interest.

To do so we place ourself in a Markovian framework by assuming that the sequence $X_1^\ell = X_1 \dots X_\ell$ is a order[2] $d \geqslant 1$ homogeneous Markov chain over the finite alphabet $\mathcal{A}$. We denote by ν its starting distribution and by π its transition matrix. For all $a_1^d \in \mathcal{A}$ and for all $b \in \mathcal{A}$ we then have: $\mathbb{P}\left(X_1^d = a_1^d\right) = \nu(a_1^d) =$ and $\mathbb{P}(X_{i+d} = b | X_i^{i+d-1} = a_1^d) = \pi(a_1^d, b)$ with $1 \leqslant i \leqslant \ell - d$.

[1] One might also think to add a fraction of one which correspond to the probability to see the corresponding letter at the degenerated position.

[2] For the sake of simplicity, the particular degenerated case where $d = 0$ is left to the reader.

For all $1 \leqslant i \leqslant \ell$ we denote by $\mathcal{X}_i \subset \mathcal{A}$ the subset of all possible values taken by X_i according to the data. For example, if we consider a IUPAC DNA sequence "ANTWY..." we have $\mathcal{X}_1 = \{\mathtt{A}\}$, $\mathcal{X}_2 = \{\mathtt{A},\mathtt{C},\mathtt{G},\mathtt{T}\}$, $\mathcal{X}_3 = \{\mathtt{T}\}$, $\mathcal{X}_4 = \{\mathtt{A},\mathtt{T}\}$, $\mathcal{X}_5 = \{\mathtt{C},\mathtt{T}\}$, ...

In a first part we establish the distribution of X_1^ℓ under the constraint that $X_1^\ell \in \mathcal{X}_1^\ell$ using an adaptation of the Baum-Welch algorithm [3]. We then demonstrate that the constrained sequence is distributed according to a heterogeneous Markov model which starting distribution and transition function have explicit expressions. This result hence allows to obtain the exact constrained distribution of a pattern by the application of known Markov chain embedding techniques. The interest of the method is finally illustrated with EST data and discussed.

2 Constrained Distribution

In order to compute the constrained probability $\mathbb{P}(X_1^d | X_1^d \in \mathcal{X}_1^d)$ we follow the sketch of the Baum-Welch algorithm [3] by introducing the Forward and Backward quantities.

Proposition 1 (Forward). *For all $x_1^\ell \in \mathcal{A}^\ell$ and $\forall i, 1 \leqslant i \leqslant \ell - d$ we define the* forward *quantity $F_i\left(x_i^{i+d-1}\right) \stackrel{def}{=} \mathbb{P}\left(X_i^{i+d-1} = x_i^{i+d-1}, X_1^{i+d-1} \in \mathcal{X}_1^{i+d-1}\right)$ which is computable by recurrence through:*

$$F_i\left(x_i^{i+d-1}\right) = \sum_{x_{i-1} \in \mathcal{X}_{i-1}} F_{i-1}\left(x_{i-1}^{i+d-2}\right) \pi\left(x_{i-1}^{i+d-2}, x_{i+d-1}\right) \tag{1}$$

for $2 \leqslant i \leqslant \ell - d + 1$ and with the initialization $F_1\left(x_1^d\right) = \nu\left(x_1^d\right) \mathbb{I}_{\mathcal{X}_1^d}(x_1^d)$ where $\mathbb{I}$ is the indicatrix function[3]*. We then obtain that:*

$$\mathbb{P}(X_1^\ell \in \mathcal{X}_1^\ell) = \sum_{x_{\ell-d}^\ell \in \mathcal{X}_{\ell-d}^\ell} F_{\ell-d}\left(x_{\ell-d}^{\ell-1}\right) \pi\left(x_{\ell-d}^{\ell-1}, x_\ell\right). \tag{2}$$

Proof. We prove Equation (1) by simply rewriting $F_i(x_i^{i+d-1})$ as:

$$F_i\left(x_i^{i+d-1}\right) = \sum_{x_{i-1} \in \mathcal{X}_{i-1}} \mathbb{P}(X_{i-1}^{i+d-1} = x_{i-1}^{i+d-1}, X_1^{i+d-1} \in \mathcal{X}_1^{i+d-1})$$

$$= \sum_{x_{i-1} \in \mathcal{X}_{i-1}} \underbrace{\mathbb{P}(X_{i-1}^{i+d-2} = x_{i-1}^{i+d-2}, X_1^{i+d-2} \in \mathcal{X}_1^{i+d-2})}_{F_{i-1}\left(x_{i-1}^{i+d-2}\right)}$$

$$\times \underbrace{\mathbb{P}(X_{i+d-1} = x_{i+d-1}, X_{i+d-1} \in \mathcal{X}_{i+d-1} | X_{i-1}^{i+d-2} = x_{i-1}^{i+d-2}, X_1^{i+d-2} \in \mathcal{X}_1^{i+d-2})}_{\pi\left(x_{i-1}^{i+d-2}, x_{i+d-1}\right) \mathbb{I}_{\mathcal{X}_{i+d-1}}(x_{i+d-1})}.$$

The proof of Equation (2) is established in a similar manner.

[3] For any set E, subset $A \subset E$ and element $a \in E$, $\mathbb{I}_A(a) = 1$ if $a \in A$ and $\mathbb{I}_A(a) = 0$ otherwise.

Proposition 2 (Backward). *For all* $x_1^\ell \in \mathcal{A}^\ell$ *and* $\forall i, 1 \leqslant i \leqslant \ell - d$ *we define the* backward *quantity* $B_i\left(x_i^{i+d-1}\right) \stackrel{def}{=} \mathbb{P}\left(X_i^\ell \in \mathcal{X}_i^\ell | X_i^{i+d-1} = x_i^{i+d-1}\right)$ *which is computable by recurrence through:*

$$B_i\left(x_i^{i+d-1}\right) = \sum_{x_{i+d} \in \mathcal{X}_{i+d}} \pi\left(x_i^{i+d-1}, x_{i+d}\right) B_{i+1}\left(x_{i+1}^{i+d}\right) \tag{3}$$

for $2 \leqslant i \leqslant \ell - d - 1$ *and with the initialization* $B_{\ell-d}\left(x_{\ell-d}^{\ell-1}\right) = \sum_{x_\ell \in \mathcal{X}_\ell} \pi\left(x_{\ell-d}^{\ell-1}, x_\ell\right) \mathbb{I}_{\mathcal{X}_{\ell-d}^{\ell-1}}(x_{\ell-d}^{\ell-1})$. *We then obtain that:*

$$\mathbb{P}(X_1^\ell \in \mathcal{X}_1^\ell) = \sum_{x_1^d \in \mathcal{X}_1^d} \nu\left(x_1^d\right) B_1\left(x_1^d\right). \tag{4}$$

Proof. The proof is very similar to the one of Proposition 1 and is hence omitted.

Theorem 1 (Marginal distributions). *For all* $x_1^\ell \in \mathcal{A}^\ell$ *we have the following results:*

a) $\mathbb{P}\left(X_1^d = x_1^d, X_1^\ell \in \mathcal{X}_1^\ell\right) = \nu\left(x_1^d\right) B_1\left(x_1^d\right)$;
b) $\mathbb{P}\left(X_i^{i+d} = x_i^{i+d}, X_1^\ell \in \mathcal{X}_1^\ell\right) = F_i\left(x_i^{i+d-1}\right) \pi\left(x_i^{i+d-1}, x_{i+d}\right) B_{i+1}\left(x_{i+1}^{i+d}\right)$;
c) $\mathbb{P}\left(X_{\ell-d}^\ell = x_{\ell-d}^\ell, X_1^\ell \in \mathcal{X}_1^\ell\right) = F_{\ell-d}\left(x_{\ell-d}^{\ell-1}\right) \pi\left(x_{\ell-d}^{\ell-1}, x_\ell\right)$;
d) $\mathbb{P}\left(X_i^{i+d-1} = x_i^{i+d-1}, X_1^\ell \in \mathcal{X}_1^\ell\right) = F_i\left(x_i^{i+d-1}\right) B_i\left(x_i^{i+d-1}\right)$.

Proof. a), b), and c) are proved using the same conditioning mechanisms used in the proofs of propositions 1 and 2. One could note that a) is a direct consequence of Equation (4), while c) could be derived from Equation (2). Thanks to Equation (3), it is also clear that b) $\Rightarrow$ d) which achieves the proof.

From now on we denote by $\mathbb{P}^C(A) \stackrel{\text{def}}{=} \mathbb{P}\left(A | X_1^\ell \in \mathcal{X}_1^\ell\right)$ the probability of an event A under the constraint that $X_1^\ell \in \mathcal{X}_1^\ell$.

Theorem 2 (Heterogeneous Markov chain). X_1^ℓ *is a order* d *heterogeneous Markov chain under* $\mathbb{P}^C$ *which starting distribution* ν^C *is given by:*

$$\nu^C(x_1^d) \stackrel{def}{=} \mathbb{P}^C\left(X_1^d = x_1^d\right) \propto \nu\left(x_1^d\right) B_1\left(x_1^d\right) \tag{5}$$

and the transition matrix π_{i+d}^C *(toward position* $i+d$*) is given by:*

$$\pi_{i+d}^C(x_1^d) \stackrel{def}{=} \mathbb{P}^C\left(X_{i+d} = x_{i+d} | X_i^{i+d-1} = x_i^{i+d-1}\right) \propto \pi\left(x_i^{i+d-1}, x_{i+d}\right) B_{i+1}\left(x_{i+1}^{i+d}\right). \tag{6}$$

Proof. Equation (5) is a direct consequence of Theorem 1a) and Equation (4). For Equation (6) we start by denoting $\mathbb{P}^C\left(X_{i+d} = x_{i+d} | X_i^{i+d-1} = x_i^{i+d-1}\right) = \mathbb{P}(A|B,C,D)$ with $A = \{X_{i+d} = x_{i+d}\}$, $B = \{X_i^{i+d-1} = x_i^{i+d-1}\}$, $C = \{X_1^i \in \mathcal{X}_1^i\}$, and $D = \{X_{i+1}^\ell \in \mathcal{X}_1^i\}$. Thanks to Bayes' formula we get that $\mathbb{P}(A|B,C,D) \propto \mathbb{P}(D|A,B,C) \times \mathbb{P}(A|B,C)$. We finally use the Markov property to get $\mathbb{P}(D|A,B,C) = B_{i+1}\left(x_{i+1}^{i+d}\right)$ and $\mathbb{P}(A|B,C) = \pi\left(x_i^{i+d-1}, x_{i+d}\right)$ which achieves the proof.

One should note the reverse sequence $X_\ell^1 = X_\ell \dots X_1$ is also a heterogeneous order d Markov model which parameters can be expressed through the Forward quantities.

3 Estimating the Background Model

Let us denote by $\theta \stackrel{\text{def}}{=} (\nu, \pi)$ the parameters of our order d Markov model. We denote by $\mathbb{P}_\theta$ all probability computation performed using the parameter θ. Since the (log-)likelihood $L\left(\theta | X_1^\ell \in \mathcal{X}_1^\ell\right) \stackrel{\text{def}}{=} \log \mathbb{P}_\theta\left(X_1^\ell \in \mathcal{X}_1^\ell\right)$ may be derived either from the Forward or Backward quantities, it is possible maximize numerically this likelihood to get the Maximum Likelihood Estimator (MLE) $\widehat{\theta} \stackrel{\text{def}}{=} \arg\max_\theta L\left(\theta | X_1^\ell \in \mathcal{X}_1^\ell\right)$.

We suggest here an alternative approach founded on the classical Expectation-Maximization algorithm for maximum likelihood estimation from incomplete data [4]. To do so, we simply consider that $X_1^\ell \in \mathcal{X}_1^\ell$ is the observed data, while $X_1^\ell = x_1^\ell$ is the unobserved data. We then get the following result:

Proposition 3 (EM algorithm). *For any starting parameter $\theta_0 \stackrel{def}{=} (\nu_0, \pi_0)$, we consider the sequence $(\theta_j)_{j \geqslant 0}$ defined for all $j \geqslant 0$ by $\theta_{j+1} \stackrel{def}{=} (\nu_{j+1}, \pi_{j+1})$ with:*

$$\nu_{j+1}\left(a_1^d\right) = \frac{\mathbb{I}_{\left\{a_1^d \in \mathcal{X}_1^d\right\}} \nu_j\left(a_1^d\right) B_1^{\theta_j}\left(a_1^d\right)}{\mathbb{P}_{\theta_j}\left(X_1^\ell \in \mathcal{X}_1^\ell\right)} \tag{7}$$

$$\pi_{j+1}\left(a_1^d, b\right) = \frac{\sum_{i=1}^{\ell-d} \mathbb{I}_{\left\{a_1^d b \in \mathcal{X}_i^{i+d}\right\}} F_i^{\theta_j}\left(a_1^d\right) \pi_j\left(a_1^d, b\right) B_{i+1}^{\theta_j}\left(a_2^d b\right)}{\sum_{i=1}^{\ell-d} \mathbb{I}_{\left\{a_1^d \in \mathcal{X}_i^{i+d-1}\right\}} F_i^{\theta_j}\left(a_1^d\right) B_i^{\theta_j}\left(a_1^d\right)} \tag{8}$$

where the $F_i^{\theta_j}$ and $B_i^{\theta_j}$ denote respectively the Forward and Backward quantities computed with the current value θ_j of the parameter, and with the convention that $B_{\ell-d+1}^{\theta_j} \equiv 1$. The sequence $(\theta_j)_{j \geqslant 0}$ converge towards a local *maximum of $L\left(\theta | X_1^\ell \in \mathcal{X}_1^\ell\right)$.*

Proof. This comes from a special application of the EM algorithm [4] where the Expectation step (Step E) consists in computing

$$Q\left(\theta | \theta_j\right) \stackrel{\text{def}}{=} \sum_{x_1^\ell \in \mathcal{X}_1^\ell} \mathbb{P}_{\theta_i}\left(X_1^\ell = x_1^\ell | X_1^\ell \in \mathcal{X}_1^\ell\right) \log \mathbb{P}_\theta\left(X_1^\ell = x_1^\ell\right)$$

while the Maximization step (Step M) consists in computing $\theta_{i+1} = \arg\max_\theta Q\left(\theta | \theta_j\right)$. Equations (7) and (8) then simply come from a natural adaptation of the classical MLE of a order d Markov chains using the pseudo counts that come directly from Theorem 1.

4 Counting Patterns

Let us consider here $\mathcal{W}$ a finite set of words over $\mathcal{A}$. We want to count the number N of positions where $\mathcal{W}$ occurs in our degenerated sequence. Unfortunately, since the sequence itself is not observed, we study instead the number N of matching positions in the random sequence X_1^ℓ under $\mathbb{P}^C$. Thanks to Theorem 2 we hence need to establish the distribution of N over a heterogeneous order d Markov chain. To do so, we perform an optimal Markov chain embedding of the problem through a Deterministic Finite Automaton (DFA) as it is suggested in [5; 6; 7; 8]. We use here the notations of [8]. Let $(\mathcal{A}, \mathcal{Q}, s, \mathcal{F}, \delta)$ be a *minimal* DFA recognizing the language[4] $\mathcal{A}^*\mathcal{W}$ of all texts over $\mathcal{A}$ ending with an occurrence of $\mathcal{W}$ (see Figure 1 for an example of such a minimal DFA). $\mathcal{Q}$ is a finite state space, $s \in \mathcal{Q}$ is the starting state, $\mathcal{F} \subset \mathcal{Q}$ is the subset of final states, and $\delta : \mathcal{Q} \times \mathcal{A} \to \mathcal{Q}$ is the transition function. We recursively extend the definition of δ over $\mathcal{Q} \times \mathcal{A}^*$ thanks to the relation $\delta(p, aw) \stackrel{\text{def}}{=} \delta(\delta(p, a), w)$ for all $p \in \mathcal{Q}, a \in \mathcal{A}, w \in \mathcal{A}^*$. We additionally suppose that this automaton is non d-ambiguous[5] which means that for all $q \in \mathcal{Q}$, $\delta^{-d}(p) \stackrel{\text{def}}{=} \left\{a_1^d \in \mathcal{A}_1^d, \exists p \in \mathcal{Q}, \delta\left(p, a_1^d\right) = q\right\}$ is either a singleton, or the empty set.

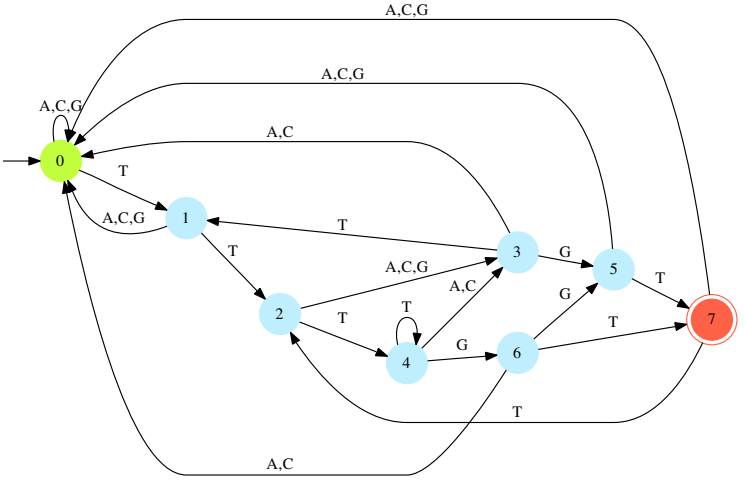

Fig. 1. Minimal DFA recognizing the language of all DNA sequences ending with an occurrence of the IUPAC pattern `TTNGT`. This DFA have a total of 8 states, $s = 0$ being the starting state, and $\mathcal{F} = \{7\}$ being the subset of final states. This DFA is 1-ambiguous since one can reach states 0 or state 3 with more than one letter.

Theorem 3 (Markov chain embedding). *We consider the random sequence over $\mathcal{Q}$ defined by $\widetilde{X}_0 \stackrel{def}{=} s$ and $\widetilde{X}_i \stackrel{def}{=} \delta(\widetilde{X}_{i-1}, X_i)$ $\forall i, 1 \leqslant i \leqslant \ell$. Under $\mathbb{P}^C$,*

[4] $\mathcal{A}^*$ denotes the set of all (possibly empty) texts over $\mathcal{A}$.
[5] A DFA having this property is also called a d-th order DFA in [7].

$(\widetilde{X}_i)_{i \geqslant d}$ is a heterogeneous order 1 Markov chain over $\mathcal{Q}' \stackrel{def}{=} \delta(s, \mathcal{A}^d \mathcal{A}^*)$ such as, for all $p, q \in \mathcal{Q}'$ and $1 \leqslant i \leqslant \ell - d$ the starting distribution $\mu_d(p) \stackrel{def}{=} \mathbb{P}^C\left(\widetilde{X}_d = p\right)$ and transition matrix $T_{i+d}(p,q) \stackrel{def}{=} \mathbb{P}^C\left(\widetilde{X}_{i+d} = q | \widetilde{X}_{i+d-1} = p\right)$ are given by:

$$\mu_d(p) = \begin{cases} \nu^C\left(a_1^d\right) & \text{if } \exists a_1^d \in \mathcal{A}^d, \delta\left(s, a_1^d\right) = p \\ 0 & \text{else} \end{cases};$$

$$T_{i+d}(p,q) = \begin{cases} \mu_{i+d}^C\left(\delta^{-d}(p), b\right) & \text{if } \exists b \in \mathcal{A}, \delta(p,b) = q \\ 0 & \text{else} \end{cases}.$$

Since Q_{i+d} contains all counting transitions, we keep track of the number of occurrences by associating a dummy variable y to these transitions. Then computing the marginal distribution at the end of the sequence would give us access to the moment generating function (mgf) of the random number of occurrences (see [5; 6; 7; 8] for more details):

Corollary 1 (Moment generating function). *The moment generating function $F(y)$ of the random number N under $\mathbb{P}^C$ is given by:*

$$F(y) \stackrel{def}{=} \sum_{k=0}^{+\infty} \mathbb{P}^C(N = k)\, y^k = \mu_d \left[\prod_{i=1}^{\ell - d} \left(P_{i+d} + y Q_{i+d}\right) \right] \mathbf{1} \tag{9}$$

where $\mathbf{1}$ is a column vector of ones and where, for all $1 \leqslant i \leqslant \ell - d$, $T_{i+d} = P_{i+d} + Q_{i+d}$ with $P_{i+d}(p,q) \stackrel{def}{=} \mathbb{I}_{q \notin \mathcal{F}} T_{i+d}(p,q)$ and $Q_{i+d}(p,q) \stackrel{def}{=} \mathbb{I}_{q \in \mathcal{F}} T_{i+d}(p,q)$ for all $p, q \in \mathcal{Q}'$.

5 Algorithm

The practical implementation of this results requires two main steps: 1) compute Forward and Backward quantities; 2) compute the mgf using Corollary 1. For the first step, the resulting complexity is $O(\ell)$ both in space and time. For the second step the space complexity is $O(D \times |\mathcal{Q}'|)$ where D is the difference between the maximum and the minimum degree of $F(y)$, and the time complexity is $O(\ell \times D \times |\mathcal{Q}'| \times |\mathcal{A}|)$ (we take here taking advantage of the sparse structure of T_{i+d}).

Using this approach on a large dataset (ex: $\ell = 5 \times 10^6$ or $\ell = 3 \times 10^9$) may then result into high memory requirements and/or long running time. Fortunately, it is possible to reduce dramatically these complexities when considering degenerated sequence where most positions are deterministic like it is the case with biological sequences.

Let us denote by $I \stackrel{\text{def}}{=} \{1 \leqslant i \leqslant \ell, |\mathcal{X}_i| > 1\}$ the set of degenerated positions in the sequence. It is clear then that the random state $\widetilde{X}_j$ is completely deterministic for all $j \in J \stackrel{\text{def}}{=} \{1 \leqslant i \leqslant \ell, j > i + d \ \forall i \in I\}$. The positions $j \in J$ thus contribute in a deterministic way to N with a fixed number of occurrence n. It

hence remains only to take into account the variable part $N - n = N_1 + \ldots + N_k$ where the N_i are independent contributions of each of the k segments of $\bar{J}$ (the complementary of J in $\{1, \ldots, \ell\}$).

If we denote by $F_i(y)$ the mgf of N_i, we get that

$$F(y) = y^n \times \prod_{i=1}^{k} F_i(y)$$

which dramatically reduces the complexity of the problem. Since each $F_i(y)$ may be obtained by a simple application of Corollary 1 on the particular (short) segment of interest, and one only need to compute the Forward-Backward quantities for this particular segment.

For example, let us consider that the observed IUPAC sequence is $x_1^\ell =$ `AAYGCANGBTAGGCTTATCWATGRT` and that $d = 2$. We have $I = \{3, 7, 9, 20, 24\}$ and $\bar{J} = [3, 5] \cup [7, 11] \cup [20, 22] \cup [24, 25]$. In order to compute $F_1(y)$, $F_2(y)$, $F_3(y)$ and $F_4(y)$, one just need to known the order $d = 2$ past before each of the corresponding segment: `AA` for the first, `CA` for the second, `TC` for the third, and `TG` for the last one.

6 Discussion

Let us consider the dataset `est_pro_01` which is described in Table 1. Here is the transition matrix over of a order $d = 1$ homogeneous Markov model over $\mathcal{A} = \{\mathtt{A}, \mathtt{C}, \mathtt{G}, \mathtt{T}\}$ estimated on this dataset using MLE (though the EM algorithm):

$$\widehat{\pi} = \begin{pmatrix} 0.3337 & 0.1706 & 0.2363 & 0.2595 \\ 0.2636 & 0.2609 & 0.1775 & 0.2980 \\ 0.2946 & 0.2218 & 0.2666 & 0.2169 \\ 0.2280 & 0.2413 & 0.2106 & 0.3201 \end{pmatrix}.$$

Since only 1% of the dataset is degenerated, we observe little difference between this rigorous estimate and one obtained through a rough heuristic (like discarding all degenerated positions in the data).

However, this result should not be taken as a rule, especially when considering more degenerated sequences (*e. g.* with 10% degenerated positions) and/or higher order Markov models (*e. g.* $d = 4$).

Using this model, it is possible to study the *observed distribution* of a pattern in the dataset by computing though Corollary 1 the distribution of its random number of occurrence N under the constrained probability $\mathbb{P}^C$. Table 2 compares the number of occurrences obtained by discarding all degenerated positions in the data (Count1) to the observed distribution. Despite the fact that only 1% of the data are degenerated, we can see that there is great differences between our naive approach and the real observed distribution. For example, if we consider the simple pattern `GCTA` we can see that the naive count of 715 occurrences lies well outside the 90% credibility interval $[727, 740]$ and we have similar results for the other considered patterns.

Table 2. Distribution of patterns in the degenerated IUPAC sequences from `est_pro_01`. Count1 is obtained by discarding all degenerated positions in the dataset, and Count2 by replacing each special letter by the most likely compatible symbol. Since the observed distribution is discrete, percentiles and median are rounded to the closest value.

pattern	Count1	Count2	min	5%-tile	median	95%-tile	max
`GCTA`	715	732	715	727	733	740	824
`TTAGT`	197	211	197	201	205	209	253
`TTNGT`	839	853	853	874	881	889	1005
`TRNANNNSTM`	472	505	477	488	493	498	535

For more complex patterns like `TTNGT` the difference between the naive count and the observed distribution is even more dramatic since 839 does not even belong to the support $[853, 1005]$ of the observed distribution. This is due to the fact that the *string* `TTNGT` actually occurs $853 - 839 = 14$ times in the dataset. Since our naive approach discards all positions in the data where a symbol other than `A`, `C`, `G` or `T` appears, these 14 occurrences are hence omitted.

If we now preprocess the dataset by replacing all degenerated symbols by the most frequent letter in the corresponding subset we get the number of occurrences denoted Count2. If this heuristic seems to give an interesting result for Pattern `GCTA` (counting close to the median), it is unfortunately not the case for the other ones for which the method results either in under-counting (Pattern `TTNGT`) or over-counting (patterns `TTAGT` and `TRNANNNSTM`).

As a general rule, it is usually difficult to predict the bias introduced by a particular heuristic since it can either lead to under- or over-coutings (for example Count1 always result in under-countings) and that this may even depend on the pattern of interest (like with Count2). The rigorous method we have here developed may hence also provide a way to test the statistical properties of a particular heuristic.

Finally, let us point out that thanks to the optimal Markov chain embedding provided by the DFA-based approach presented above, we are here able to deal with relatively complex patterns like `TRNANNNSTM`.

7 Conclusion

In this paper, we provide a rigorous way to deal with the distribution of Markov chains over a finite alphabet $\mathcal{A}$ under the constraint that each position X_i of the sequence belongs to restricted subset $\mathcal{X}_i \subset \mathcal{A}$. We provide a Forward-Backward framework to compute marginal distributions and derive from it a EM estimation procedure. We also prove that the resulting constrained distribution is a heterogeneous Markov chains and provide explicit formulas to recursively compute its transition matrix. Thanks to this result, it is possible to apply known DFA-based methods from pattern theory to study the distribution of a pattern of interest in this constrained sequence, hence providing a trustful observed distribution for

the pattern number of occurrences. This information may then be used to derive a p-value p for a pattern by combining p_n the p-value of the observation of n occurrences in a unconstrained dataset with the observed distribution through formulas like $p = \sum_n p_n \mathbb{P}^C(N = n)$.

One should note that the approach we introduce here may have more applications than just counting patterns in IUPAC sequences. For example, one might use a similar approach to take into account the occurrences positions of known patterns of interest thus allowing to derive distribution of patterns conditionally to a possibly complex set of other patterns. One should also point out that the constraint $X_i \in \mathcal{X}_i$ should easily be complexificated, for example by considering a specific distribution over $\mathcal{X}_i$. For instance, such a distribution may come from the posterior decoding probabilities of a sequencing machine.

From the computational point of view, it is essential to understand that the heterogeneous nature of the Markov chain we consider forbid to use classical computational tricks like power computations. The resulting complexity is hence linear with the sequence length ℓ rather that logarithmic. However, one should expect a dramatic improvement of the method by restricting the use of heterogeneous Markov models only in the vicinity of degenerated positions like it is suggested in Section 5. With such an approach, one might rely on classical pattern matching for 99% of the data, and the method presented above would be restricted to the study of the 1% remaining data. Using this computational trick, it hence seems possible to rely on the rigorous exact computation introduced here rather than on a bias heuristic.

Finally, we have demonstrated with our example that even a small amount of degenerated data may have huge consequences in terms of pattern frequencies, and thus possibly affect every subsequent analysis method involving these frequencies like Markov and hidden Markov model estimations and pattern studies. Considering the possible bias caused by degenerated letters in biological data, and the reasonable complexity of the exact solution we introduce in this paper, our study suggests that the problem of degenerated data in pattern related analysis should no longer be ignored.

References

[1] IUPAC: International Union of Pure and Applied Chemistry (2009), `http://www.iupac.org`

[2] EMBL: European Molecular Biology Laboratory Nucleotide Sequence Database (2009), `http://www.ebi.ac.uk/embl/`

[3] Baum, L.E., Petrie, T., Soules, G., Weiss, N.: A maximization technique occurring in the statistical analysis of probabilistic functions of markov chains. Ann. Math. Statist. 41(1), 164–171 (1970)

[4] Dempster, A.P., Laird, N.M., Rubin, D.B.: Maximum Likelihood from Incomplete Data via the EM Algorithm. Journal of the Royal Stat. Society. Series B 39(1), 1–38 (1977)

[5] Nicodème, P., Salvy, B., Flajolet, P.: Motif statistics. Theoretical Com. Sci. 287(2), 593–617 (2002)

[6] Crochemore, M., Stefanov, V.: Waiting time and complexity for matching patterns with automata. Info. Proc. Letters 87(3), 119–125 (2003)
[7] Lladser, M.E.: Mininal markov chain embeddings of pattern problems. In: Information Theory and Applications Workshop, pp. 251–255 (2007)
[8] Nuel, G.: Pattern markov chains: optimal markov chain embedding through deterministic finite automata. J. of Applied Prob. 45(1), 226–243 (2008)