Revisiting Waiting Times in DNA Evolution

Pierre Nicodeme

Revisiting Waiting Times in DNA Evolution[*]

Pierre Nicodème

LIPN - Team CALIN, CNRS-UMR 7030, University Paris North,
Institut Galilée , 99, Avenue Jean-Baptiste Clément, 93430 - Villetaneuse, France
phone: +33 (0)14940-4069, fax: +33 (0)14826-0712, pierre.nicodeme@lipn.univ-paris13.fr.

January 20, 2014

Abstract

Transcription factors are short stretches of DNA (or k-mers) mainly located in promoters sequences that enhance or repress gene expression. With respect to an initial distribution of letters on the DNA alphabet, Behrens and Vingron [3] consider a random sequence of length n that does not contain a given k-mer or word of size k. Under an evolution model of the DNA, they compute the probability $\mathfrak{p}_n$ that this k-mer appears after a unit time of 20 years. They prove that the waiting time for the first apparition of the k-mer is well approximated by $T_n = 1/\mathfrak{p}_n$. Their work relies on the simplifying assumption that the k-mer is not self-overlapping. They observe in particular that the waiting time is mostly driven by the initial distribution of letters. Behrens *et al.* [2] use an approach by automata that relaxes the assumption related to words overlaps. Their numerical evaluations confirms the validity of Behrens and Vingron approach for non self-overlapping words, but provides up to 44% corrections for highly self-overlapping words such as `AAAAA`. We devised an approach of the problem by clump analysis and generating functions; this approach leads to prove a quasi-linear behaviour of $\mathfrak{p}_n$ for a large range of values of n, an important result for DNA evolution. We present here this clump analysis, by language decomposition, and by an automaton construction; finally, we describe an equivalent approach by construction of Markov automata.

1 Introduction

Several theoretical studies have tried to give a probabilistic explanation for the speed of changes in transcriptional gene regulation (e.g. [14], [4]).

[*] This work received support of the ANR project MAGNUM number ANR BLAN- 0204 07

Behrens and Vingron [3] infer how long one has to wait until a given Transcription Factor (TF for short) binding site emerges at random in a promoter sequence. Using a Bernoulli probabilistic model denoted by M0 and estimating evolutionary substitution rates based on multiple species promoter alignments for the three species *Homo sapiens*, *Pan troglodytes* and *Macaca mulatta*, they compute the expected waiting time for every k-mer, k ranging from 5 to 10, until it appears in a human promoter. They conclude that the waiting time for a TF binding site is highly determined by its composition and that indeed TF binding sites can appear rapidly, i.e. in a time span below the speciation time of human and chimp.

However, in their approach, Behrens and Vingron [3] rely on the assumption that if a k-mer of interest appears more than once in a promoter sequence, it does not overlap with itself. This particularly affects the waiting times for highly autocorrelated words like e.g. `AAAAA` or `CTCTCTCTCT`. Using automata, Behrens *et al.* [2] relaxed this assumption and, thus, more accurately compute the expected waiting times until appearance of k-mers.

While Behrens and Vingron [3] and all preceding works were mostly interested in sequences of fixed length $n = 1000$, Behrens *et al.* [2] realized experimentally that $\mathfrak{p}_n$ behaves asymptotically linearly with n for a wide range of lengths. This linear behaviour was next proved by the author of the present article; this property is biologically important, since the lengths of promoters are in an approximate range from 1000 base pairs to 2000 base pairs; moreover, it cannot be deduced easily from the rigorous computations by automata of Behrens *et al.* [2]. We give here proofs of this property that are based on clump analysis and use either combinatorics on words or automata constructions, and singularity analysis of the resulting generating functions. Our adaptation of previous methods is new and has theoretical and practical interests.

We present the model in Section 2. We recall in Section 3 the Behrens-Vingron equations (2010) and the automaton approach of Behrens *et al.* (2012). The main part of the article is devoted in Section 4 to counting the number $\widetilde{H}_n$ of putative-hit positions in random sequences of length n, assuming that they do not contain the targeted k-mer b; at first order, the probability $\mathfrak{p}_n$ is then a linear function of $\widetilde{H}_n$. We provide in this section the background for the Guibas-Odlyzko language decomposition and its extension to clump analysis, and a parallel construction by automata. This section also contains the translation to generating functions of the formal languages used and states our result of quasi-linearity of $\mathfrak{p}_n$; the proof of this result is given in Section A of the Appendix. Section 5 sketches a proof by automaton that does not rely on clump constructions.

A) Estimations for $\nu(a)$, $a \in \mathcal{A}$:

$\nu(A)$	$\nu(C)$	$\nu(G)$	$\nu(T)$
0.23889	0.26242	0.25865	0.24004

B) Estimations for $p_{\alpha,\beta}(1)$, $\alpha, \beta \in \mathcal{A}$:

	A	C	G	T
A	9.99999996e-01	4.54999995e-09	1.57499996e-08	3.40000002e-09
C	6.14999993e-09	9.99999996e-01	7.14999985e-09	2.17499994e-08
G	2.17499994e-08	7.14999985e-09	9.99999996e-01	6.14999993e-09
T	3.40000002e-09	1.57499996e-08	4.54999995e-09	9.99999998e-01

Table 1: **Parameter estimations.** Numbers taken from [3].

2 Preliminaries

Throughout the article, we assume that promoter sequences evolve according to model M0 which has been described by [3].

Model M0. Given an alphabet $\mathcal{A} = \{\mathtt{A}, \mathtt{C}, \mathtt{G}, \mathtt{T}\}$, let $S(0) = (S_1(0), \ldots, S_n(0))$ denote the initial promoter sequence of length n taking values in this alphabet. We assume that the letters in $S(0)$ are independent and identically distributed with $\nu(x) := \Pr(S_1(0) = x)$. Let the time evolution $(S(t))_{t\geq 0}$ of the promoter sequence be governed by the 4×4 infinitesimal rate matrix $\mathbb{Q} = (r_{\alpha,\beta})_{\alpha,\beta\in\mathcal{A}}$. The matrix $\mathbb{P}(t) = (p_{\alpha,\beta}(t))_{\alpha,\beta\in\mathcal{A}}$ containing the transitions probabilities of α evolving into β in finite time $t \geq 0$, $(\alpha, \beta \in \mathcal{A})$, can be computed by $\mathbb{P}(t) = e^{t\mathbb{Q}}$; see Karlin-Taylor [8], p. 150-152. Table 1 provides the parameters used.

The expected waiting time. Given a binding site

$$b = (b_1, \ldots, b_k) \quad \text{where } b_1, \ldots, b_k \in \mathcal{A}, \tag{1}$$

the aim is to determine the expected waiting time until b emerges in a promoter sequence of length n provided that it does not appear in the initial promoter sequence $S(0)$. More precisely, let

$$\begin{aligned} T_n = \inf\{t \in \mathbb{N} : \quad & \exists i \in \{1, \ldots, n-k+1\} \\ & \text{such that } (S_i(t), \ldots, S_{i+k-1}(t)) = (b_1, \ldots, b_k)\}. \end{aligned}$$

Then, given that Pr(b occurs in $S(0)$) = 0, the waiting time T_n has approximately a geometric distribution with parameter $\mathfrak{p}_n$ verifying

$$\begin{aligned}\mathfrak{p}_n &= \Pr(b \text{ occurs in generation } 1 \mid b \text{ does not occur in generation } 0) \qquad (2)\\ &= \Pr(b \in S(1) \mid b \notin S(0)).\end{aligned}$$

See [3]. In particular, one has $\mathbf{E}(T_n) \approx \dfrac{1}{\mathfrak{p}_n}$.

3 Previous work

We present briefly Behrens and Vingron [3] and Behrens *et al.* [2] methods.

3.1 Behrens-Vingron (2010)

Considering the k-mer $b = b_1 \dots b_k$, Behrens and Vingron consider (i) the probability that it occurs at time $t = 1$ in $S(1)$ and (ii) the probability that it evolves from a k-mer of $S(0)$. Case (i) is computed by inclusion-exclusion and by assuming that the word b is not self-overlapping. This gives [1]

$$Pr(|S(1)|_b \geq 1) = \sum_{\ell=1}^{\lfloor n/k \rfloor} (-1)^{\ell+1} \binom{n-(k-1)l}{l} \Pr(|S(1)|_b = \ell).$$

Taking in account the evolution probability, they consider next the words at substitution distance 1 to k of b. Assuming that the insertion of such words within a sequence $S(0)$ with no occurrence of b does not create an occurrence of b (this is wrong in general, but a good approximation for non-overlapping words long enough in a 4 letters alphabet), they obtain

$$\Pr\left(|S(1)|_b \geq 1 \;\middle|\; |S(0)|_b = 0 \right) \approx \sum_{\ell=1}^{\lfloor n/k \rfloor} (-1)^{\ell+1} \binom{n-(k-1)l}{l} p_\ell, \qquad (3)$$

$$\text{with} \quad p_\ell = \left(\sum_{(a_1 \dots a_k) \in \mathcal{A}^k \setminus \{b_1 \dots b_k\}} \nu(a_1) \dots \nu(a_k) \prod_{i=1}^{k} p_{a_i, b_i}(1) \right)^l. \qquad (4)$$

In the last equation, p_ℓ is the approximate probability that b occurs ℓ times in $S(1)$ while not occurring in $S(0)$.

[1] We use the notation $|w|_u$ to note the number of occurrences of a word u in a word w.

	BNN		BV		
	$\mathbf{E}_{\mathrm{BNN}}(T_{1000})/10^6$	Rank	$\mathbf{E}_{\mathrm{BV}}(T_{1000})/10^6$	Rank	$\frac{\mathbf{E}_{\mathrm{BNN}}(T_{1000})}{\mathbf{E}_{\mathrm{BV}}(T_{1000})}$
CCCCC	9.105	1021	6.304	1	1.44
GGGGG	9.570	1022	6.666	142	1.44
TTTTT	10.401	1023	7.457	993	1.39
AAAAA	10.656	1024	7.654	1024	1.39
CGCGC	7.047	699	6.446	11	1.09
TCCCC	7.076	737	6.477	17	1.09
CCCCT	7.076	738	6.477	21	1.09
GCGCG	7.127	787	6.518	31	1.09
CTCTC	7.263	883	6.679	148	1.09
CACAC	7.337	945	6.750	217	1.09

Table 2: **Expected waiting times (generations) for 5-mers in model M0 with $\frac{\mathbf{E}_{\mathrm{BNN}}(T_{1000})}{\mathbf{E}_{\mathrm{BV}}(T_{1000})} > 1.09$.** (Top 10 results from Table 2 of Behrens *et al.* [2]). $\mathbf{E}_{\mathrm{BV}}(T_{1000})$ denotes the expected waiting time according to Behrens-Vingron [3] (BV) and $\mathbf{E}_{\mathrm{BNN}}(T_{1000})$ according to the automaton approach of Behrens *et al.* [2] (BNN). Ranks refer to 5-mers sorted by their waiting time of appearance according to the two different procedures BV and BNN; rank 1 is assigned to the fastest evolving 5-mer, rank 1024 (=4^5) to the slowest emerging 5-mer.

3.2 Automaton approach of Behrens *et al.* (2012)

Behrens *et al.* [2] use the following algorithm. Let $\mathsf{A}_b = (Q := \{0, 1, \ldots, k\}, \delta_b, 0, \{k\})$ be the Knuth-Morris-Pratt automaton over the alphabet $\mathcal{A}$ that recognizes the language $\mathcal{A}^\star b \mathcal{A}^\star$. The language $\mathcal{A}^\star \setminus \mathcal{A}^\star b \mathcal{A}^\star$ is recognized by the automaton $\overline{\mathsf{A}}_b = (Q, \delta_b, 0, \{0, 1, \ldots, k-1\})$. They construct a product automaton $\mathsf{P} = \overline{\mathsf{A}}_b \otimes \mathsf{A}_b$ on $\mathcal{A} \times \mathcal{A}$ such that

$$\mathsf{P} = (Q \times Q, \delta, (0,0), F), \text{ with } \begin{cases} \delta((p,q),(\alpha,\beta)) = (\delta_b(p,\alpha), \delta_b(q,\beta)) \\ F = \{0, 1, \ldots, k-1\} \times \{k\}. \end{cases}$$

They weight (i) any transition $q \xrightarrow{a} q'$ of $\overline{\mathsf{A}}_b$ by $\nu(a)$ and (ii) any transition $c \xrightarrow{(a,a')} c'$ of P by $\nu(a) \times p_{a \to a'}$, where ν is the initial distribution of letters and $p_{x \to y}$ is the probability of evolution of letter x to letter y in a unit time. Considering the corresponding adjacency matrices $\overline{\mathbb{A}}_b$ and $\mathbb{P}$, (provided a suitable reordering of the lines and columns of the matrices), V_F being a column vector with a one for each final state in P and zero elsewhere, the probability $\mathfrak{p}_n$ verifies,

$$\mathfrak{p}_n = (1, 0, \ldots, 0) \times \mathbb{P}^n \times V_F \Big/ (1, 0, \ldots, 0) \times \overline{\mathbb{A}}_b^n \times (\overbrace{1, \ldots, 1}^{k \text{ times}}, 0)^t.$$

Table 2 provides the top 10 5-mers with respect with the correction done by Behrens *et al.* (2012) with respect to Behrens-Vingron (2010).

Considering the minimal period $m(b)$ of a k-mer b, such that

$$m(b) = \min(i, \ |u| = i; \quad b = u^i.v, \ v \text{ prefix of } u),$$

and noting i-periodic a word with minimal period i, half of the 5-mers, two-thirds of the 7-mers and all of the 10-mers with $\frac{\mathbf{E}_{\mathrm{BNN}}(T_{1000})}{\mathbf{E}_{\mathrm{BV}}(T_{1000})} > 1.05$ are either 1- or 2-periodic, i.e. show a high degree of autocorrelation. This implies that, for only 4% of the 5-mers, 0.2% of the 7-mers and 0.002% of the 10-mers, the exact computations of Behrens *et al.* (2012) differ by more than 5% of the approximate computations of Behrens-Vingron (2010). However, as shown in Behrens *et al.* (2012), a non negligible number of Transcription Factors are highly correlated.

4 Clump approach

Table 2 shows clearly the importance of autocorrelation.

Assuming a four letters alphabet with a uniform probability distribution, founding an occurrence of `AAAAA` at a position, up to boundary effects, we have a probability 1/4 of finding an occurrence shifted by one position. In contrast, considering an occurrence of `AACCC`, we need reading at least 5 new letters to find a new occurrence, and the probability of finding two consecutive occurrences is $1/4^5$. This is a well known fact in combinatorics of words; words occur by clumps and, while clumps of a non-overlapping word have only one occurrence of the word, clumps of an overlapping word may have several; since the probability (in a uniform model) of occurrence of any word of a given size at any position is the same, the proportion of text covered by clumps of a non-overlapping word will be larger than that of a self-overlapping word. This property extends to sets of words depending of their self-overlap structure.

We show here that the number of positions in $S(0)$ that can mutate and provide an occurrence of a k-mer b in $S(1)$, or putative-hit positions, is neither a function of the number of occurrences of b in $S(1)$ nor of the number of occurrences of the neighbours of b in $S(0)$, but that this number can be computed by a variant of clump analysis.

Notations. Given a word b, we note $d(b)$ the set of its neighbours at edit distance 1 (by substitution of one letter), and $d_\ell(b)$ the vector resulting of a lexicographic sort of $d(b)$. Therefore, for an alphabet $\mathcal{A} = \{\mathtt{A}, \mathtt{C}\}$, we have $d(\mathtt{ACC}) = \{\mathtt{CCC}, \mathtt{AAC}, \mathtt{ACA}\}$ and $d_\ell(\mathtt{ACC}) = (\mathtt{AAC}, \mathtt{ACA}, \mathtt{CCC})$.
The correlation set $\mathcal{C}_{v_1,v_2}$ of two words v_1 and v_2 is defined as usual,

$$\mathcal{C}_{v_1,v_2} = \{\ e \ \mid \text{ there exists } e' \in \mathcal{A}^+ \text{ such that } v_1 e = e' v_2 \text{ with } |e| < |v_2|\ \}.$$

When we have $w = v_1 = v_2$, we get $\mathcal{C}_{w,w} = \mathcal{C}$ (the autocorrelation of w).

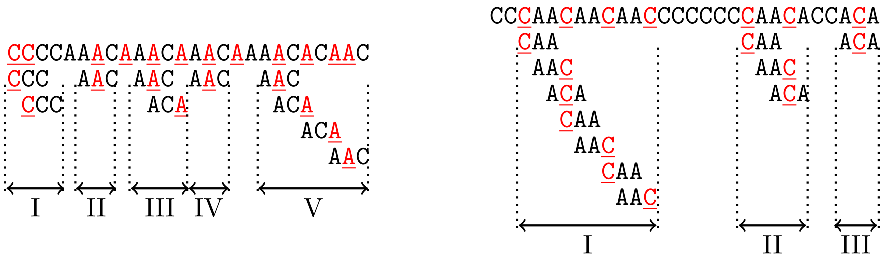

Figure 1: **Clumps and putative-hit positions.** Sequences $S_b(0)$ for $b = \texttt{ACC}$ (left) and $S_{b'}(0)$ for $b' = \texttt{AAA}$ (right). The sequence $S_b(0)$ (resp. $S_{b'}(0)$) avoids the k-mer b (resp. b'). Putative-hit positions are underlined and in red. Clumps are shown at their respective positions under the sequences. Note that extensions to the right of clumps of the set $d(\texttt{AAA})$ for $b' = \texttt{AAA}$, while creating a new occurrence of a word of the set, do not add necessarily a new putative-hit position; clump I (right) contains 7 occurrences of $d(\texttt{AAA})$, but only 4 putative-hit positions for $b' = \texttt{AAA}$. Therefore the number of word occurrences is not the relevant statistics for precisely counting putative-hit positions. Note also in the clump I for $b = \texttt{ACC}$ (left) that, when the right extension of a clump adds a new putative-hit position, this position is not necessarily in the extension, but possibly backwards left.

Putative-hit positions. Given a sequence $S(0)$ not containing a k-mer b, a putative-hit position is any position of $S(0)$ that can lead by a mutation to an occurrence of b in $S(1)$, where we assume that a single mutation has occurred. We have for instance

$$S(0) = \texttt{CCCAACAC}, \quad b = \texttt{ACC} \qquad \rightsquigarrow \qquad \underline{S}(0) = \underline{\texttt{C}}\texttt{CCA}\underline{\texttt{A}}\texttt{CAC},$$

where the putative-hit positions are underlined in $\underline{S}(0)$. Mutating any single putative-hit position of $\underline{S}(0)$ leads to a sequence $S(1)$ with an occurrence of $b = \texttt{ACC}$. Examples of sequences $S(0)$ for the 3-mers $\texttt{ACC}$ and $\texttt{AAA}$ (see Figure 1) reveal that the right method to carry on the computation of putative-hit positions is clump analysis [1].

Aim of the computation. We assume during the mainstream of our proofs that there is a unique type of mutation, and we generalize in the Appendix. In the following, H_n is the random variable counting the number of putative-hit positions for a word b in a random sequence $S(0)$ of length n. We consider the generating function $F_b(z, t)$ that counts the number of putative-hit positions for the k-mer b in texts avoiding this k-mer[2]

$$F_b(z,t) = \sum_{w \in \widehat{\mathcal{A}_b^\star}} \Pr(w) t^{\text{put-hit-pos}(w)} z^{|w|} = \sum_{n \geq 0} \sum_{w \in \widehat{\mathcal{A}_b^n}} \Pr(w) t^{H_n} z^n, \qquad (5)$$

where put-hit-pos(w) is the number of putative-hit positions of the k-mer b in the word w. Note that, up to probability of second order small magnitude, only one putative-hit position will mutate.

Assuming that we can compute $F_b(z,t)$, we obtain next

$$F_b(z,1) = \widehat{\mathcal{A}_b^\star}(z,1) = \sum_{n \geq 0} \widehat{f}_n^{(b)} z^n = \sum_{n \geq 0} \Pr(S_n(0) \notin \mathcal{A}^\star b \mathcal{A}^\star) \times z^n, \qquad (6)$$

where $\widehat{f}_n^{(b)}$ is the probability that a random sequence of length n does not contain the word b. We prove in the Appendix that for a constant θ, we have $\mathfrak{p}_n \approx \theta \times \mathbf{E}(\widetilde{H}_n)$, where the conditionned[3] expectation $\mathbf{E}(\widetilde{H}_n)$ verifies

$$\mathbf{E}(\widetilde{H}_n) = \mathbf{E}(H_n) \Big/ \widehat{f}_n^{(b)} = [z^n] \left. \frac{\partial F_b(z,t)}{\partial t} \right|_{t=1} \Big/ \widehat{f}_n^{(b)}. \qquad (7)$$

4.1 Analysis “à la Guibas-Odlyzko”

Considering a reduced set of words (no word is factor of another word in the set), Régnier and Szpankowski [12, 13] and Régnier [11] use (as an evolution of Guibas and Odlyzko previous work [6, 7]) a natural parsing or decomposition of texts with respect to the occurrences of the set.

We follow here the corresponding presentation of Lothaire [9] (Chapter 7). Let $\mathcal{V} = \{v_1, \ldots, v_r\}$ be a reduced set of words. We have, formally

Definition 4.1 *Right, Minimal, Ultimate and Zero-Occ languages.*

- *The “Right” language $\mathcal{R}_i$ associated to the word v_i is the set of words $\mathcal{R}_i = \{r \mid r = e \cdot v_i$ and there is no $v \in \mathcal{V}$ such that $r = xvy$ with $|y| > 0\}$.*
- *The “Minimal” language $\mathcal{M}_{ij}$ leading from a word v_i to a word v_j is the set of words $\mathcal{M}_{ij} = \{m \mid v_i \cdot m = e \cdot v_j$ and there is no $v \in \mathcal{V}$ such that $v_i \cdot m = xvy$ with $|x| > 0, |y| > 0\}$.*
- *The “Ultimate” language of words following the last occurrence of the word v_i (such that this occurrence is the last occurrence of $\mathcal{V}$ in the text) is the set of words $\mathcal{U}_i = \{u \mid$ there is no $v \in \mathcal{V}$ such that $v_i \cdot u = xvy$ with $|x| > 0\}$.*
- *The “Zero-Occ” language $\mathcal{Z}$ is the set of words with no occurrences of $\mathcal{V}$, $\mathcal{Z} = \{w \mid$ there is no $v \in \mathcal{V}$ such that $w = xvy\}$.*

It is possible to obtain the generating functions of these languages by combinatorics and by new automata constructions.

[2] For a language $\mathcal{L}$ and the k-mer b, we note $\widehat{\mathcal{L}_b}$ (or simply $\widehat{\mathcal{L}}$) the maximal subset of $\mathcal{L}$ that has no occurrence of b, $\widehat{\mathcal{L}} = \big\{w \in \mathcal{L}; \quad |w|_b = 0\big\}$.

[3] We use the classical equation $\Pr(A|B) = \Pr(A)/\Pr(B)$ for two events A and B such that $B \subset A$.

4.2 Régnier-Szpankowski equations

Considering the matrix $\mathbb{M} = (\mathcal{M}_{ij})$ and using $\mathcal{C}_{ij} = \mathcal{C}_{v_i,v_j}$ as a shorthand, we have

$$\bigcup_{k\geq 1}\left(\mathbb{M}^k\right)_{i,j} = \mathcal{A}^\star\cdot v_j + \mathcal{C}_{ij} - \delta_{ij}\epsilon, \qquad \mathcal{U}_i\cdot\mathcal{A} = \bigcup_j \mathcal{M}_{ij} + \mathcal{U}_i - \epsilon, \tag{8}$$

$$\mathcal{A}\cdot\mathcal{R}_j - (\mathcal{R}_j - v_j) = \bigcup_i v_i\mathcal{M}_{ij}, \qquad \mathcal{Z}\cdot v_j = \mathcal{R}_j + \bigcup_i \mathcal{R}_i\left(\mathcal{C}_{ij} - \delta_{ij}\epsilon\right), \tag{9}$$

where the Kronecker symbol δ_{ij} is 1 if $i = j$ and 0 elsewhere. These equations are non-ambiguous and translate to generating functions, where for a language $\mathcal{L}$ and its generating function $L(z)$, we have $L(z) = \sum_{w\in\mathcal{L}} \mathrm{Pr}(w)z^{|w|}$. Translating the system of Equations (6, 7) to generating functions and solving the resulting system provide the generating functions $R_i(z)$, $M_{i,j}(z)$, $U_j(z)$ and $Z(z)$ of the Right, Minimal, Ultimate and Zero-Occ languages. The parsing by languages is now reflected in the following equation

$$\frac{1}{1-z} = Z(z) + (R_1(z), \dots, R_r(z))(\mathbb{I} - \mathbb{M}(z))^{-1}\begin{pmatrix} U_1(z) \\ \vdots \\ U_r(z) \end{pmatrix} \tag{10}$$

where $\frac{1}{1-z}$ is the generating function of $\mathcal{A}^\star$, the set of all texts.

4.3 Automata constructions for a reduced set $\mathcal{V} = \{v_1, \dots, v_r\}$

The languages $\mathcal{R}_i, \mathcal{M}_{ij}, \mathcal{U}_j$ and $\mathcal{Z}$ are recognized by the following new automata constructions (where $\bigotimes$ is the usual automaton product):

$$\mathcal{R}_i = \mathcal{A}^\star.v_i \bigotimes_{s\in\{1,\dots,r\}} \mathrm{Not}(\mathcal{A}^\star v_s\mathcal{A}) \qquad v_i\mathcal{M}_{ij} = v_i\mathcal{A}^\star \bigotimes \mathcal{A}^\star.v_j \bigotimes_{s\in\{1,\dots,r\}} \mathrm{Not}(\mathcal{A}^+ v_s\mathcal{A}^+)$$

$$v_j\mathcal{U}_j = v_j\mathcal{A}^\star \bigotimes_{s\in\{1,\dots,r\}} \mathrm{Not}(\mathcal{A}^+ v_s\mathcal{A}^\star) \qquad \mathcal{Z} = \mathrm{Not}\left(\bigotimes_{s\in\{1,\dots,r\}} \mathcal{A}^\star v_s\mathcal{A}^\star\right)$$

4.4 Constrained languages

Language approach. Considering a word b, with $|b| = r$, and such that $d_\ell(b) = (v_1, \dots, v_r)$, we can compute from the vector of words $(v_1, \dots, v_r, b)$ a row vector of Right languages $(\mathcal{R}_1, \dots, \mathcal{R}_r, \mathcal{R}_{r+1})$, a matrix of Minimal languages $(\mathcal{M}_{ij})$ with i and j from 1 to $r+1$, a column vector of Ultimate languages $(\mathcal{U}_1,\dots,\mathcal{U}_r,\mathcal{U}_{r+1})^{\mathbf{t}}$ and the language $\mathcal{Z}$. Extracting the languages with indices from 1 to r provides us for the k-mer b with the Right $\widehat{\mathcal{R}}_i = \mathcal{R}_i$, Minimal $\widehat{\mathcal{M}}_{ij} = \mathcal{M}_{ij}$ and Ultimate $\widehat{\mathcal{U}}_j = \mathcal{U}_j$ languages [4] avoiding this k-mer.

Automata approach. It is also immediate to construct by automata the constrained languages. For instance, we have, for $i,j \in \{1,\dots,r\}$,

$$v_i\widehat{\mathcal{M}_{ij}} = v_i\mathcal{A}^\star \bigotimes \mathcal{A}^\star.v_j \bigotimes_{s\in\{1,\dots,r\}} \mathrm{Not}(\mathcal{A}^+ v_s \mathcal{A}^+) \bigotimes \mathrm{Not}(\mathcal{A}^\star b \mathcal{A}^\star).$$

4.5 Clump equations by language decomposition

Bassino *et al.* [1] modify the Régnier-Szpankowski analysis of reduced sets to more specifically consider clumps of occurrences, where a clump is constituted either (*i*) of a single isolated (with no overlap with other occurrences) occurrence of a word of the pattern, or (*ii*) of a maximal set of occurrences where each occurrence overlaps at least another one.

We consider the *residual* language $\mathcal{D} = \mathcal{L}.w^-$ as $\mathcal{D} = \{v,\ v\cdot w \in \mathcal{L}\}$. Considering two languages $\mathcal{L}_1$ and $\mathcal{L}_2$, we write $\mathcal{L}_2 - \mathcal{L}_1 = \mathcal{L}_2 \setminus \mathcal{L}_1 = \{v;\ v \in \mathcal{L}_2, v \notin \mathcal{L}_1\}$.

The clumps can be generated by a matrix of codes $\mathbb{K} = (\mathcal{K}_{ij})$. With

$$\mathcal{K}_{ij} = \mathcal{B}_{ij} - \mathcal{B}_{ij}\mathcal{A}^+ \quad \text{and} \quad \left\{ \begin{array}{l} \mathcal{B}_{ij} = \mathcal{C}_{ij} \cap \mathcal{M}_{ij} \quad \text{if } i \neq j, \\ \mathcal{B}_{ii} = (\mathcal{C}_{ii} - \epsilon) \cap \mathcal{M}_{ii}, \end{array} \right. \tag{11}$$

the language decomposition by clumps for a pattern $\mathcal{V} = \{v_1,\dots,v_r\}$ is

$$\mathcal{A}^\star = \mathcal{Z} + (\mathcal{R}_1 v_1^-,\dots,\mathcal{R}_r v_r^-)\,\mathbb{G}\Big((\mathbb{M}-\mathbb{K})^-\mathbb{G}\Big)^\star \begin{pmatrix} \mathcal{U}_1 \\ \vdots \\ \mathcal{U}_r \end{pmatrix}, \text{ with } \left\{ \begin{array}{l} \mathbb{K} = (\mathcal{K}_{ij}), \quad \mathbb{S} = \mathbb{K}^\star, \\ \mathbb{G} = \big(v_i \mathbb{S}_{ij}\big) \end{array} \right. \tag{12}$$

Example 4.2 *For the word $w = \texttt{AAAA}$, we have $\mathcal{C} = \{\epsilon, \texttt{A}, \texttt{AA}, \texttt{AAA}\}$ and $\mathcal{K} = \{\texttt{A}\}$. For the pattern $\mathcal{V} = \{\texttt{TATAT}, \texttt{CATAT}\}$, we have $\mathcal{C}_{\texttt{CATAT,TATAT}} = \{\texttt{AT}, \texttt{ATAT}\}$ and $\mathcal{K}_{\texttt{CATAT,TATAT}} = \{\texttt{AT}\}$. For the pattern $\mathcal{V}' = \{\texttt{CAA}, \texttt{AAT}, \texttt{AAA}\}$, we have $\mathcal{C}_{\texttt{CAA,AAT}} = \{\texttt{T}, \texttt{AT}\}$ and $\mathcal{K}_{\texttt{CAA,AAT}} = \{\texttt{T}\}$.*

[4] Note that $\widehat{\mathcal{Z}} = \mathcal{Z}$ and $\widehat{\mathcal{A}_b^\star} = \widehat{\mathcal{Z}} + (\widehat{\mathcal{R}_1},\dots,\widehat{\mathcal{R}_r})\Big(\widehat{\mathcal{M}_{ij}}\Big)(\widehat{\mathcal{U}_1},\dots,\widehat{\mathcal{U}_r})^t$.

Constrained clumps. The finite code languages generating the correlation languages of two words are easy to compute directly; one must however also avoid the forbidden word b while extending clumps. We therefore define for v_i (resp. v_j) the i-th (resp. j-th) entry of the sequence $d_\ell(b)$

$$\widehat{\mathcal{K}}_{ij} = \{h \in \mathcal{K}_{ij}; \qquad |v_i.h|_b = 0\},$$

where $|g|_b$ is again the number of occurrences of the word b in the word g. Since the code sets $\mathcal{K}_{ij}$ are finite, the computations of the sets $\widehat{\mathcal{K}}_{ij}$ can be done with finite complexity $\kappa \leq \sum_{i,j} \sum_{h \in \mathcal{K}_{ij}} |v_i.h|$.

Gathering everything, we obtain a constrained version of Equation (10) for the language $\widehat{\mathcal{A}_b^\star}$ of texts avoiding the word b,

$$\widehat{\mathcal{A}_b^\star} = \widehat{\mathcal{Z}} + (\widehat{\mathcal{R}}_1 v_1^-, \dots, \widehat{\mathcal{R}}_r v_r^-)\; \widehat{\mathbb{G}}\Big((\widehat{\mathbb{M}} - \widehat{\mathbb{K}})^- \widehat{\mathbb{G}}\Big)^\star \begin{pmatrix} \widehat{\mathcal{U}_1} \\ \vdots \\ \widehat{\mathcal{U}_r} \end{pmatrix}, \text{ with } \begin{cases} \widehat{\mathbb{K}} = (\widehat{\mathcal{K}}_{ij}), \\ \widehat{\mathbb{S}} = \widehat{\mathbb{K}}^\star, \\ \widehat{\mathbb{G}} = \Big(v_i \widehat{\mathbb{S}}_{ij}\Big) \end{cases} \tag{13}$$

4.6 Computing the generating function of the number of putative-hit positions

We prove that the computation of the generating function $F_b(z,t)$ of Equation (5) follows from Equation (11). Indeed, taking in consideration the lengths of the words and the number of occurrences of putative-hit positions (that are occurrences of $v_i \in d(b)$), we have first $v_i(z,t) = \Pr(v_i) t z^{|v_i|}$ for each $v_i \in d(b)$. Next, for each $\widehat{\mathcal{K}}_{ij}$, we can compute by string matching the number of putative-hit positions in each word of $v_i.\widehat{\mathcal{K}}_{ij}$. This gives

$$\widehat{\mathcal{K}}_{ij}(z,t) = \sum_{w \in \widehat{\mathcal{K}}_{ij}} \Pr(w) t^{\text{put-hit-pos}(v_i.w)-1} z^{|w|},$$

where we substracted the putative-hit position occurring within v_i.

From the last equation and Equation (11), we get

$$\begin{aligned} &\widehat{\mathbb{K}}(z,t) = \Big(\widehat{\mathcal{K}}_{ij}(z,t)\Big), \quad \widehat{\mathbb{S}}(z,t) = \Big(\mathbb{I} - \widehat{\mathbb{K}}(z,t)\Big)^{-1}, \\ &\widehat{\mathbb{G}}(z,t) = \Big(v_i(z,t)\widehat{\mathbb{S}}_{ij}(z,t)\Big). \end{aligned} \tag{14}$$

Substituting in Equation (11) $\widehat{\mathbb{G}}$ by $\widehat{\mathbb{G}}(z,t)$ and $\widehat{\mathcal{Z}}, \widehat{\mathcal{R}}_i v_i^-, (\widehat{\mathbb{M}}–\widehat{\mathbb{K}})$ and $\widehat{\mathcal{U}}_i$ with $1 \leq i \leq r$ by their translations to generating functions (that depend only of the variable z) provides the expression of $F_b(z,t)$ that has been formally defined in Equation(5).

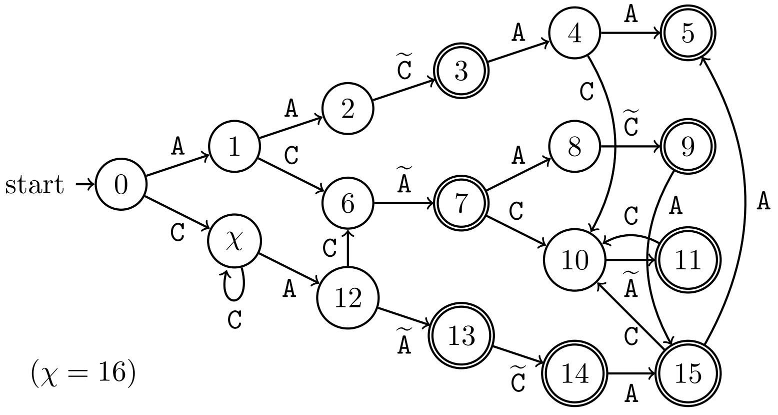

Figure 2: Automaton for constrained clumps of $d(\texttt{AAA}) = \{\texttt{AAC}, \texttt{ACA}, \texttt{CAA}\}$. Double circles signals an occurrence of one of these words. Transitions covered by tildes $(\widetilde{\texttt{A}}, \widetilde{\texttt{C}})$ emits a signal counting a putative-hit position. The missing transitions A have been erased since we want to avoid occurrences of $b = \texttt{AAA}$. The missing transitions C point to the state χ. All states are terminal.

Considering again the evolution matrix $\mathbb{P}(1) = (p_{\alpha\to\beta})$ with $\alpha, \beta \in \{\texttt{A}, \texttt{C}, \texttt{G}, \texttt{T}\}$, and using Equation (14), we state the following proposition that we prove in the Appendix, Section A.

Proposition 4.3 *For (i)* $\max_{\alpha,\beta\in\mathcal{A};\alpha\neq\beta}(p_{\alpha\to\beta}) \ll 1$ *and (ii)* n *large enough with* $n \ll \min^{-1}_{\alpha,\beta\in\mathcal{A};\alpha\neq\beta}(p_{\alpha\to\beta})$*, the probability* $\mathfrak{p}_n$ *that a* k*-mer occurs at time* 1 *while not occuring at time* 0 *in a random sequence of length* n *behaves quasi-linearly with respect to the length* n*. The convergence to this quasi-linear regime is exponential.*

4.7 Approach by automata of clumps

We can alternatively use the construction of clumps by automata given in Bassino *et al.* [1].

For a set $\mathcal{V} = \{v_1, \dots, v_r\}$ with correlation sets $\mathcal{C}_{ij}$ we construct a kind of "Aho-Corasick" automaton on the following set of words X

$$X = \{v_i \cdot w \mid 1 \leq i \leq r \text{ and } w \in \{\epsilon\} \cup \mathcal{C}_{ij} \text{ for some } j\}.$$

The considered automaton T is built on X with set of states $Q = \mathrm{Pref}(X)$ and start or initial state $s = \epsilon$. The transition function is defined (as in the Aho-Corasick construction) by

$$\delta(p, x) = \quad \text{the longest suffix of } px \in \mathrm{Pref}(X).$$

We build with this construction, for any k-mer b, the automaton recognizing the clumps of the neighbours $d(b)$ of b while avoiding occurrences of b; this last condition can be made effective by doing an automaton product. Assuming that the set of states of the resulting automaton T is $Q = \{0, 1, \dots, s\}$ and that the initial state is labelled 0, we set all the states of the automaton T to terminal to recognize all sequences avoiding b. Therefore, we have

$$\mathsf{T} = (\{0, 1, \dots, s\}, \delta, 0, \{0, 1, \dots, s\}).$$

See Figure 2 for an example with the alphabet $\mathcal{A} = \{\mathtt{A}, \mathtt{C}\}$, the k-mer $b = \mathtt{AAA}$ and $d(b) = \{\mathtt{AAC}, \mathtt{ACA}, \mathtt{CAA}\}$. Transitions with a “tilde” correspond to finding a new putative-hit position in the last occurrence of a word of $d(b)$ that has been read.

Clump-Core. We consider the set of states O that recognize an occurrence of $d(b)$,

$$O = \{q, \quad \delta(0, w) = q, \ w \in X\}.$$

We also consider the set of states $\overline{E}$ that do not belong to a clump extension,

$$\overline{E} = \{q, \quad \delta(0, w) = q, \ w \in \widehat{\mathrm{Pref}}(d(b))\},$$

where $\widehat{\mathrm{Pref}}(d(b))$ is the set of strict prefixes of words of $d(b)$.

We define finally the Clump-Core of the automaton by its set of states E which verifies

$$E = Q \setminus \overline{E}.$$

Referring to the automaton of Figure 2, we have $\overline{E} = \{0, 1, 2, 16\,(\chi), 6, 12\}$ and $E = \{3, 4, 5, 7, 8, 9, 10, 11, 13, 14, 15\}$.

Markov property. By construction of the automaton, for any word w with $|w| \leq |b|$, we have the following property,

$$\forall e \in E, \ \forall w \text{ with } (|w| \leq |b|) \begin{cases} \nexists w' \neq w \text{ with } (|w'| = |w|) \\ \text{such that } \delta(q_1, w) = \delta(q_2, w') = e. \end{cases}$$

This property can be proved iteratively with respect to the length of the words.

Handling the putative-hit positions. For simplicity, we assume that there is only one type of mutation, but the method extends to the general case. We count as previously the putative-hit positions by the variable t.

For each state $o \in O$ (recognizing an occurrence of $d(b)$), let $\theta(o)$ be the word w with $|w| \leq |b|$, of maximal length, and verifying,

1. there exists q such that $\delta(q, w) = o$,

2. there is no $u \in \widehat{\mathrm{Pref}}(w)$ such that $\delta(q, u) \in O$.

By the Markovian property, this defines a unique word. Referring to Figure 2, we have $\theta(7) = \mathtt{ACA}$, $\theta(5) = \mathtt{AA}$, $\theta(14) = \mathtt{C}$, and $\theta(15) = \mathtt{A}$. Moreover, the Markovian property asserts that reading backward $|b|$ transitions from any state $o \in O$ does a reverse spelling of a unique word of $d(b)$. We can next locate any putative-hit position within this word and check if it belongs to $\theta(o)$.

The adjacency matrix $\mathbb{H}(t) = (h_{ij}(t))$ associated to the automaton T is then defined as follows: $h_{ij}(t) = 0$ if there is no transition from i to j; elsewhere, let us assume that $\delta(i, \alpha) = j$. We have then

$$h_{ij}(t) = \begin{cases} \Pr(\alpha) \text{ if } \left| \begin{array}{l} j \notin O, \\ j \in O \text{ and } \theta(j) \text{ contains no putative-hit position,} \end{array} \right. \\ \Pr(\alpha) \times t \text{ elsewhere.} \end{cases}$$

The generating function $F_b(z, t)$ defined in Equation (5) verifies

$$\begin{aligned} F_b(z, t) &= (1, 0, \ldots, 0) \times \big(\mathbb{I} + z\mathbb{H}(t) + \cdots + z^n \mathbb{H}^n(t) + \ldots\big) \times \mathbf{1}^t \\ &= (1, 0, \ldots, 0) \times (\mathbb{I} - z\mathbb{H}(t))^{-1} \times \mathbf{1}^t. \end{aligned}$$

5 Yet another proof by automata

We sketch a proof that does not make use of clumps. The construction is computationally very costly.

We build the (pruned) Knuth-Morris-Pratt automaton K recognizing $\overline{\mathcal{A}^\star b \mathcal{A}^\star}$ (the set of sequences avoiding the k-mer b).

Next we compute the order-$(2|b|-1)$ Markov automaton M of K. The transitions of this automaton are words of size $2|b|$. It is possible by reading the transitions to know when a new putative-hit position is present, and to multiply the corresponding entry in the associated adjacency matrix by the counting variable t. Let $\mathbb{M}(t)$ be this matrix. The matrix associated to the automaton K is positive, irreducible and transitive; so is the matrix $\mathbb{M}(t)$, disregarding a trie-like structure leading to its recurrent part. Writing p_{mut} the probability of mutation, we can make the substitution $t \rightsquigarrow (1-p_{\mathrm{mut}}) + x \times p_{\mathrm{mut}}$. We then have for the recurrent part $\mathbb{R}(t)$ of $\mathbb{M}(t)$,

$$\mathbb{U}(x) := \mathbb{R}((1-p_{\mathrm{mut}}) + x \times p_{\mathrm{mut}}) = \mathbb{Y} + xp_{\mathrm{mut}}\mathbb{X}.$$

Assuming that $n \times p_{\mathrm{mut}} = o(1)$, we get for a polynomial $P(x)$

$$\mathbb{U}^n(x) = \mathbb{Y}^n + xnp_{\mathrm{mut}}\mathbb{Y}^{n-1}\mathbb{X} + x^2P(x) \times O\left((np_{\mathrm{mut}})^2\right). \tag{15}$$

Writing ξ_u and ξ_y the dominant eigenvalues of $\mathbb{U}(1)$ and $\mathbb{Y}$, the property $np_{\mathrm{mut}} = o(1)$ entails that $\xi_r^n = \xi_y^n \times (1+o(1))$. We then deduce from Equation (15) that

$$\mathfrak{p}_n \approx \frac{[x^1](1,0,\ldots,0)\mathbb{U}^n(x)\mathbf{1}^t}{(1,0,\ldots,0)\mathbb{U}^n(1)\mathbf{1}^t} = (\alpha + \beta \times n) \times (1+o(1)).$$

6 Conclusion

We provided several methods for analysing waiting times in DNA evolution that give insights in the structure of the problem. We showed that clump analysis and generating functions are powerful and convenient tools for this aim and used either combinatorics on words or automata constructions. In particular we proved the property of quasi-linearity related to the probability of first occurrence of a k-mer after a unit time.

Acknowledgements. We thank Sarah Behrens and Cyril Nicaud for helpful discussions and technical help.

References

[1] F. BASSINO, J. CLÉMENT, J. FAYOLLE, P. NICODÈME, Constructions for Clump Statistics. In: P. JACQUET (ed.), *Proceedings of the Fifth Colloquium on Mathematics and Computer Science, Blaubeuren, Germany.* DMTCS, 2008, 183–198.
http://www-lipn.univ-paris13.fr/~bassino/publications/mathinfo08.pdf.

[2] S. BEHRENS, C. NICAUD, P. NICODÈME, An automaton approach for waiting times in DNA evolution. *Journal of Computational Biology* 19 (2012) 5, 550–562.

[3] S. BEHRENS, M. VINGRON, Studying the evolution of promoters: a waiting time problem. *J. Comput. Biol* 17 (2010) 2, 1591–1606.
http://www.liebertonline.com/doi/full/10.1089/cmb.2010.0084.

[4] R. DURRETT, D. SCHMIDT, Waiting for regulatory sequences to appear. *Ann. Appl. Probab.* 17 (2007) 1, 1–32.

[5] P. FLAJOLET, R. SEDGEWICK, *Analytic Combinatorics*. Cambridge University Press, 2009.

[6] L. GUIBAS, A. ODLYZKO, Periods in strings. *J. Combin. Theory* A (1981) 30, 19–42.

[7] L. GUIBAS, A. ODLYZKO, Strings Overlaps, Pattern Matching, and Nontransitive Games. *J. Combin. Theory* A (1981) 30, 108–203.

[8] S. KARLIN, H. TAYLOR, *A First Course in Stochastic Processes*. Academic Press, 1975. Second Edition, 557 pages.

[9] M. LOTHAIRE, *Applied Combinatorics on Words*. Encyclopedia of Mathematics, Cambridge University Press, 2005.

[10] P. NICODÈME, B. SALVY, P. FLAJOLET, Motif Statistics. *Theoretical Computer Science* 287 (2002) 2, 593–618.

[11] M. RÉGNIER, A Unified Approach to Word Occurrences Probabilities. *Discrete Applied Mathematics, Special issue on Computational Biology* 104 (2000) 1, 259–280.

[12] M. RÉGNIER, W. SZPANKOWSKI, On the Approximate Pattern Occurrences in a Text. In: *SEQUENCES '97: Proceedings of the Compression and Complexity of Sequences 1997*. IEEE Computer Society, Washington, DC, USA, 1997, 253.

[13] M. RÉGNIER, W. SZPANKOWSKI, On Pattern Frequency Occurrences in a Markovian Sequence. *Algorithmica* 22 (1998) 4, 631–649.

[14] J. R. STONE, G. A. WRAY, Rapid evolution of cis-regulatory sequences via local point mutations. *Mol. Biol. Evol.* 18 (2001) 9, 1764–1770.

A Singularity analysis

The methods developed in Section 4.6 apply to any k-mer with any finite alphabet. Moreover, using the additivity of expectations, we could split the putative-hit positions along their types; with the toy alphabet $\{\mathtt{A},\mathtt{C}\}$, we would get two putative-hit positions type, $(\mathtt{A}\to\mathtt{C})$ and $(\mathtt{C}\to\mathtt{A})$. By following the same footsteps as in Section 4.6, we can now compute the expectations of putative-hit positions $\mathbf{E}\left(H_n^{(\mathtt{A}\to\mathtt{C})}\right)$ and $\mathbf{E}\left(H_n^{(\mathtt{C}\to\mathtt{A})}\right)$ which correspond to the two types of mutation. Considering only the case $(\mathtt{A}\to\mathtt{C})$, we can by pattern-matching compute $\widehat{\mathcal{K}}_{ij}\left(z,t_{(\mathtt{A}\to\mathtt{C})}\right)$. We have as previously $v_i\left(z,t_{(\mathtt{A}\to\mathtt{C})}\right)=\Pr(v_i)z^{|v_i|}t_{(\mathtt{A}\to\mathtt{C})}$.

We write in the following for sake of simplicity $x=t_{(\mathtt{A}\to\mathtt{C})}$, and consider the generating function $F_b(z,x)=F_b(z,t_{(\mathtt{A}\to\mathtt{C})})$ where the function $F_b(z,t)$ is defined in Equation (5).

The solutions of the Régnier-Szpankowski equations provide functions that are rational [5]. Similarly, each term of the matrix Equation (11) is rational and so are the

[5] This property follows also from an equivalent approach by finite automata and use of the Chomsky-Schützenberger algorithm [10] that leads to solve a linear system of equations.

corresponding extensions to counts of putative-hit positions that lead to the explicit value of $F_b(z,x)$.

We can therefore write for two polynomials $P(z,x)$ and $Q(z,x)$

$$F_b(z,x) = \frac{P(z,x)}{Q(z,x)} \quad \text{and} \quad F_b(z,1) = \sum_{n\geq 0} \widehat{f}_n^{(b)} z^n = \frac{P(z,1)}{Q(z,1)},$$

where, again, $\widehat{f}_n^{(b)}$ is the probability that a random sequence of length n has no occurrence of b. We have

$$E(z) = \sum_{n\geq 0} \mathbf{E}\left(H_n^{(\mathtt{A}\to\mathtt{C})}\right) z^n = \frac{\partial}{\partial x} F_b(z,x)\bigg|_{x=1} = \frac{P'_x(z,1)}{Q(z,1)} - \frac{P(z,1)Q'_x(z,1)}{Q^2(z,1)}.$$

This series has only positive coefficients and by Pringsheim Theorem [5][Th. IV.6, p.240], it has a real positive singularity on the circle of convergence that we note τ; by considering the automaton recognizing $\overline{\mathcal{A}^\star b \mathcal{A}^\star}$, the associated irreducible and primitive matrix, and Perron-Frobenius properties of positive matrices [8], this real positive singularity is dominant. The singularity τ is also the smallest positive solution of $Q(z,1) = 0$.

Therefore, extracting the nth Taylor coefficient of the generating functions $E(z)$ and $F_b(z,1)$ by Cauchy integrals along a circle of radius $\tau < R < \tau_2$, where τ_2 is the value of the second largest singularity(ies) in modulus, we obtain for constants ψ, ϕ_1 and ϕ_2

$$\widehat{f}_n^{(b)} = \psi \times \tau^{-(n-1)} \left(1 + \mathcal{O}\left(B^n\right)\right), \qquad (B < 1),$$

$$\text{and} \qquad \mathbf{E}(H_n^{(\mathtt{A}\to\mathtt{C})}) = [z^n]E(z) = \tau^{-n}(\phi_1 \times n + \phi_2) \times \left(1 + \mathcal{O}\left(B^n\right)\right). \tag{16}$$

It follows then immediately that

$$\mathbf{E}\left(\widetilde{H}_n^{(\mathtt{A}\to\mathtt{C})}\right) = \mathbf{E}(H_n^{(\mathtt{A}\to\mathtt{C})}) \Big/ \widehat{f}_n = (c_1 \times n + c_2) \times \left(1 + \mathcal{O}\left(B^n\right)\right), \qquad (B < 1). \tag{17}$$

In the more general case, we have, for $n \ll \min^{-1}_{\alpha,\beta\in\mathcal{A}} \left(p_{\alpha\to\beta}\right)$,

$$\mathfrak{p}_n \approx \sum_{\substack{\alpha\in\mathcal{A},\beta\in\mathcal{A}\\ \alpha\neq\beta}} \mathbf{E}\left(\widetilde{H}_n^{(\alpha\to\beta)}\right) \times p_{\alpha\to\beta}(1) = (C_1 \times n + C_2) \times \left(1 + \mathcal{O}\left(K^n\right)\right), \quad (K < 1),$$

where C_1 and C_2 are constants, and K is the maximum of the $|\mathcal{A}|(|\mathcal{A}| - 1)$ constants B used when applying the Equation (17) to the $|\mathcal{A}|(|\mathcal{A}| - 1)$ types of mutation.

This proves Proposition 4.3.

B Toy example for the clump approach by language

We consider the following toy example

$$\mathcal{A} = \{\mathtt{A}, \mathtt{C}\}, \quad b = \mathtt{ACAC}, \quad b' = \mathtt{AACC}, \qquad \Pr(\mathtt{A}) = \Pr(\mathtt{C}) = \frac{1}{2}.$$

We want to estimate the expectations of the *total* number of putative-hit positions ($\mathtt{A}\to\mathtt{C}$) and ($\mathtt{C}\to\mathtt{A}$) for the words b and b'.

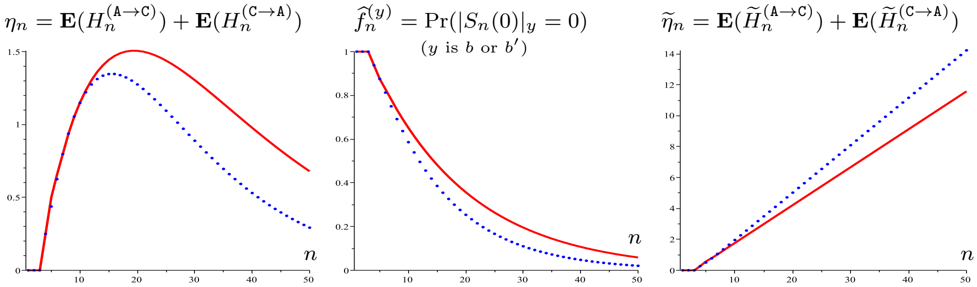

Figure 3: Asymptotic linear behaviour of the unconditionned η_n (left) and conditionned $\widetilde{\eta}_n = \eta_n / \widehat{f}_n^{(y)}$ (right) expectations of the number of putative-hit positions for $b = \texttt{ACAC}$ (plain red lines) and $b' = \texttt{AACC}$ (blue dots) with the alphabet $\{\texttt{A}, \texttt{C}\}$ and $\Pr(\texttt{A}) = \Pr(\texttt{C}) = 1/2$. See Equations (13), (16) and (17).

Equation (5) becomes, with $\widehat{\mathcal{A}_b^n}$ the subset of sequences of size n of $\widehat{\mathcal{A}_b^\star}$,

$$F_b(z,t) = \sum_{n\geq 0} \sum_{w \in \widehat{\mathcal{A}_b^n}} \frac{z^n}{2^n} t^{\text{put-hit-pos}(w)} \tag{18}$$

As mentioned earlier, putative-hit positions only occur in the clumps, and therefore the core of differences between the behaviour of the 4-mers $b = \texttt{ACAC}$ and $b' = \texttt{AACC}$ come from differences in the matrices of codes $\widehat{\mathbb{K}_b}$ and $\widehat{\mathbb{K}_{b'}}$.

We have

$b = \texttt{ACAC}$
$d_\ell(b) = (\texttt{AAAC}, \texttt{ACAA}, \texttt{ACCC}, \texttt{CCAC})$

$$\widehat{\mathbb{K}_b} = \begin{pmatrix} 0 & \frac{z^2 t}{4} & \frac{z^2 t}{4} & \frac{z^3 t}{8} \\ \frac{z^3 t}{8} + \frac{z^2}{2} & \frac{z^3 t}{8} & \frac{z^3 t}{8} & 0 \\ 0 & 0 & 0 & \frac{z^3 t}{8} + \frac{z^2}{2} \\ 0 & \frac{z^2 t}{4} & \frac{z^2 t}{4} & \frac{z^3 t}{8} \end{pmatrix}$$

$b' = \texttt{AACC}$
$d_\ell(b') = (\texttt{AAAC}, \texttt{AACA}, \texttt{ACCC}, \texttt{CACC})$

$$\widehat{\mathbb{K}_{b'}} = \begin{pmatrix} 0 & \frac{zt}{2} & 0 & 0 \\ \frac{z^3 t}{8} & \frac{z^3 t}{8} & 0 & \frac{z^2 t}{4} \\ 0 & 0 & 0 & \frac{z^3 t}{8} \\ 0 & 0 & \frac{zt}{2} & \frac{z^3 t}{8} \end{pmatrix}$$

$$d_\ell(b)(z,t) = d_\ell(b')(z,t) = (\tfrac{z^4 t}{16}, \tfrac{z^4 t}{16}, \tfrac{z^4 t}{16}, \tfrac{z^4 t}{16})$$

The last equations [6] intuitively suggests that there should be more putative-hit positions in a random sequence for $b = \texttt{ACAC}$ than for $b' = \texttt{AACC}$. This is verified in Figure 3 (left) where we plot the unconditionned expectations of the number of putative-hit positions for both 4-mers. However, when conditionning as in Figure 3 (right), the 4-mer $\texttt{ACAC}$ gets lower expectations than the 4-mer $\texttt{AACC}$; this follows from the values of the constants C_1 for b and b' that respectively are $C_1 = 0.2452503893$ for $b = \texttt{ACAC}$ and $C_1 = 0.3068491678$ for $b' = \texttt{AACC}$.

Figure 3 moreover exhibits the linear behaviour of these expectations with respect to the length n of the sequences, as stated in Proposition 4.3.

[6] The extension $\texttt{AC} \in \widehat{\mathcal{K}}_{\texttt{ACAA},\texttt{AAAC}}$ as in $\texttt{ACA}\underline{\texttt{A}}|\texttt{AC}$ leads to *no new* putative-hit position. The same remark applies to the extension $\texttt{AC} \in \widehat{\mathcal{K}}_{\texttt{ACCC},\texttt{CCAC}}$.