Assessing the Significance of Sets of Words

Valentina Boeva[1], Julien Clément[2],
Mireille Régnier[3], and Mathias Vandenbogaert[4]

[1] Moscow State University, Vorob'evy Gory, Russia
valeyo@mail333.com
[2] Igm, Université de Marne-la-Vallée, France
Julien.Clement@univ-mlv.fr
[3] Inria, 78153 Le Chesnay, France
Mireille.Regnier@inria.fr
[4] Biozentrum, Basel Universitat, Switzerland
mathias.vandenbogaert@unibas.ch

Abstract. Various criteria have been defined to evaluate the significance of sets of words, the computation of them often being difficult. We provide explicit expressions for the waiting time in such a context. In order to assess the significance of a cluster of potential binding sites, we extend them to the co-occurrence problem. We point out that these criteria values depend on a few fundamental parameters. We provide efficient algorithms to compute them, that rely on a combinatorial interpretation of the formulae. We show that our results are very tight in the so-called twilight zone and improve on previous rough approximations. One assumes that the text is generated according to a Markov stationary process. These results are developed for an extended model of consensus.

1 Introduction

Many statistical softwares have been designed in the last decade to search for exceptional words and predict biological functions *In silico*. Using some pattern matching algorithms to detect some candidates, one assesses their biological significance by comparing the observed number and the expected number. The main differences between softwares are the underlying probability models, the searching algorithms and the comparison criteria. One common underlying assumption is that the genome is randomly generated according to some probability model. In this paper, the model can be either Bernoulli or Markov. The comparison criteria depend on the applications and the parameters of the problem. One first classification arises from the *size* of the problem, that is to say the number of word occurrences and the text size. A typical application with large texts is the search of recognition sites for Restriction Modification Systems in a genome. It is shown in [1, 2] that these sites are avoided words. Symmetrically, it is proved in [3] that the *Chi*-motif is overrepresented in *E. Coli*. One may also search for common words in a set of (small) sequences. A typical application is the search of regulatory signals in upstream sequences of genes that are either orthologous or coregulated. When the signal is degenerated, each signal can be represented

A. Apostolico, M. Crochemore, and K. Park (Eds.): CPM 2005, LNCS 3537, pp. 358–370, 2005.

by a set $\mathcal{H}_1$ or $\mathcal{H}_2$. This set is defined from experimental data, as a consensus or a position matrix (PSSM, PWM...).

Word counting results have been derived by several authors for a single word H or a set of words $\mathcal{H}$. Recently, special attention has been paid to the simultaneous occurrence of different binding sites in upstream sequences. Indeed, it has been observed [4] in eucaryotes that multiple transcription factors binding to the same transcription control region are often involved in the same transcriptional regulation. Hence, the co-occurrence and spatial relationships of individual binding sites provides a better understanding of regulation and of multifactorial control of gene expression. Therefore, we formalize below the co-occurrence problem and extend previous exact formulae in this case. A second aim of this paper is the rewriting of exact matricial expressions [5, 6] or induction algorithms [7, 8] in a suitable form. Indeed, we define a few fundamental parameters and show that tight numerical computations depend on these parameters. Third, simple combinatorial interpretations as overlapping sequences are provided for exact expressions or fundamental parameters. We provide efficient algorithms to compute them. In passing, we define an extended consensus model that is close to PSSM (Positional Specific Scoring Matrix) and suitable for the search of regulatory signals.

2 Main Steps and Results

Distribution formulae or algorithms always involve the possible overlaps of the words to be counted [5, 6, 9, 10]. The correlation sets were introduced in the seed paper [11]. We define the notion of *complement* and *minimal complement*.

Definition 1. *Given two strings* F *and* G*, the* overlap set *of* F *and* G *is the set of suffixes of* F *that are proper prefixes of* G*. Any suffix of* G *in the associated factorizations of* G *is named a* right complement *of* F *in* G*. The set of* right complements *of* F *in* G *is called the* correlation set *of* F *and* G *and denoted* $C_{\mathrm{F,G}}$*. When* $\mathrm{F} = \mathrm{G}$*, the* autocorrelation set *is* $A_{\mathrm{F}} = C_{\mathrm{F,F}} + \varepsilon$ *with* ε *the empty word.*

Given a set $\mathcal{H}$*, a right complement of a word* F *in* $\mathcal{H}$ *is any right complement of* F *in a word* G *in* $\mathcal{H}$*. A right complement* w *of* F *is* minimal *iff no proper prefix of* w *is a right complement of* F *in* $\mathcal{H}$*. The set of* minimal right complements *of* F *that belong to* $C_{\mathrm{F,G}}$ *is denoted* $\widetilde{C}_{\mathrm{F,G}}$*.*

We assume below that all words have the same size m. The next definition provides tools for word counting.

Definition 2. *Given a set of* q *words* $\mathcal{H} = (\mathrm{H}_i)_{1 \le i \le q}$*, one denotes* $H(z) = (P(\mathrm{H}_1)z^m, \ldots, P(\mathrm{H}_q)z^m)$*. The* probability matrix $\mathbb{H}(z)$ *is the* $q \times q$ *matrix with* q *rows equal to* $H(z)$*. Given two words* H_i *and* H_j*, the* complement polynomial *(resp.* complement minimal polynomial*) is*

$$C_{i,j}(z) = \sum_{w \in C_{\mathrm{H}_i,\mathrm{H}_j}} P(w) z^{|w|} \quad \textit{(resp. } \widetilde{C}_{i,j}(z) = \sum_{w \in \widetilde{C}_{\mathrm{H}_i,\mathrm{H}_j}} P(w) z^{|w|}\textit{).}$$

The complement matrix *(resp.* complement minimal matrix*) is the* $q \times q$ *matrix* $\mathbb{C}(z) = (C_{i,j}(z))_{1 \le i,j \le q}$ *(resp.* $\widetilde{\mathbb{C}}(z) = (\widetilde{C}_{i,j}(z))_{1 \le i,j \le q}$*) and the* correlation matrix *is* $\mathbb{A}(z) = \mathbb{I} + \mathbb{C}(z)$. *The* fundamental counting matrix *is*

$$\mathbb{D}(z) = (1 - z)\mathbb{A}(z) + \mathbb{H}(z).$$

When one counts a single word, $\mathbb{D}(z)$ reduces to a polynomial $D(z)$ in the Bernoulli case [11] or a series in a Markov model [5]. The following theorem, where the *fundamental counting matrix* plays a central rôle, is stated for the Bernoulli case in [5, 12] and for the Markov case in [6].

Theorem 1. *Let* $R(z) = \sum_n R_n(\mathcal{H}) z^n$ *with* $R_n(\mathcal{H})$ *the probability that the first occurrence of a word from* $\mathcal{H}$ *ends at position* n. *The generating function satisfies*

$$R(z) = H(z)\ \mathbb{D}(z)^{-1}\ \mathbf{1}_q{}^{\mathrm{t}}, \tag{1}$$

where $\mathbf{1}_q$ *a row vector with* q *columns equal to 1. Let* $t_n^k(\mathcal{H})$ *be the probability that* k *occurrences of a word from a set* $\mathcal{H}$ *occur in a text of size* n. *The generating function* $T_k(z) = \sum_{n \ge 0} t_n^k(\mathcal{H}) z^n$ *satisfies*

$$T_k(z) = H(z)\mathbb{D}(z)^{-1}\mathbb{M}(z)^{k-1}\mathbb{D}(z)^{-1}\mathbf{1}_q{}^{\mathrm{t}},$$

where $\mathbb{M}(z)$ *is the* minimal matrix *defined as* $\mathbb{M}(z) = \mathbb{I} + (z - 1)\mathbb{D}(z)^{-1}$.

Here, we unify the two definitions of the polynomial $D(z)$ and the fundamental counting matrix. Indeed, the waiting time depends on a *fundamental multioccurrence series*, that we define below.

Definition 3. *The* fundamental multioccurrence series *of a set* $\mathcal{H}$ *is*

$$Q_{\mathcal{H}}(z) = \mathrm{Trace}\left(\mathbb{H}(z)\mathbb{A}^{-1}(z)\right).$$

When $\mathcal{H}$ reduces to a single word H, one has $D(z) = (1 - z + Q(z))A(z)$.

In this paper, we first extend Theorem 1 for several sets of words, in order to address the co-occurrence problem. Counting results are also rewritten as some functions of the fundamental multioccurrence series. Simple combinatorial interpretations as overlapping sequences are provided for these functions. Hence, it turns out that the so-called z-scores, and our tight numerical approximations for the waiting time as well, depend on a few fundamental parameters. Namely,

Definition 4. *Given a set of words* $\mathcal{H}$, *one denotes* $P(\mathcal{H}) = \sum_{\mathrm{F} \in \mathcal{H}} P(\mathrm{F})$,

$$C(\mathcal{H}) = \sum_{\mathrm{F},\mathrm{G} \in \mathcal{H}} \sum_{w \in C_{\mathrm{F},\mathrm{G}}} P(\mathrm{F}w),\ \widetilde{C}(\mathcal{H}) = \sum_{\mathrm{F},\mathrm{G} \in \mathcal{H}} \sum_{w \in \widetilde{C}_{\mathrm{F},\mathrm{G}}} P(\mathrm{F}w).$$

where $P(\mathcal{H})$ *is called the* occurrence probability, $C(\mathcal{H})$ *is called the* overlap factor *of* $\mathcal{H}$ *and* $\widetilde{C}(\mathcal{H})$ *is called the* minimal overlap factor.

Note that $P(\mathcal{H}) = \mathrm{Trace}(\mathbb{H}(1))$. Overlap factors $C(\mathcal{H})$ and $\widetilde{C}(\mathcal{H})$ are the sum of the coefficients of $\mathbb{H}(1)\mathbb{C}(1)$ and $\mathbb{H}(1)\widetilde{\mathbb{C}}(1)$, respectively.

Statistical criteria in computational biology can be expressed as simple functions of parameters $P(\mathcal{H})$ and $C(\mathcal{H})$. Let $O_n(\mathcal{H})$ be the number of occurrences of words with overlap from $\mathcal{H}$ in a random text of size n under a Bernoulli model. The mean $\mathrm{E}[O_n(\mathcal{H})]$ and the variance $\mathrm{Var}[O_n(\mathcal{H})]$ satisfy [13]

$$\begin{aligned}\mathrm{E}[O_n(\mathcal{H})] &= (n-m+1)\,P(\mathcal{H}),\\ \mathrm{Var}[O_n(\mathcal{H})] &= (n-m+1)\big(\,P(\mathcal{H}) + (1-2m)\,P(\mathcal{H})^2 + 2C(\mathcal{H})\big)\\ &\quad + m(m-1)\,P(\mathcal{H})^2 - 2\widehat{C}(\mathcal{H}),\end{aligned}$$

where $\widehat{C}(\mathcal{H}) = \sum_{\mathrm{F},\mathrm{G}\in\mathcal{H}} \sum_{w\in A_{\mathrm{F},\mathrm{G}}} |w| \times P(\mathrm{F}w)$ is a slight modification of $C(\mathcal{H})$. We will show in Section 4 how to compute these quantities in an efficient way.

A correcting factor is derived for a single pattern in the Markov model in [5] and extended for a set of patterns in [6]. As this factor mainly depends on the stationary distribution, it can be viewed as a *preprocessing*.

These results allow for an efficient computation of the z-score in computational biology. When k occurrences of $\mathcal{H}$ are observed in a sequence of length n, the z-score is $Z(\mathcal{H}) = \frac{k-\mathrm{E}[O_n(\mathcal{H})]}{\sqrt{\mathrm{Var}[O_n(\mathcal{H})]}}$. This is an empirical measure of the departure from the normal distribution. High values of $|Z(\mathcal{H})|$ indicate an over representation (positive values) or an underrepresentation (negative values).

3 Waiting Time

We address below different variants of the waiting time problem. Indeed, the meaningful event may be the apparition – or not – of a word in the sequence under study. We consider also the co-occurrence problem.

3.1 Waiting Time Generating Functions

An analytic expression of the probability of first occurrence is derived in [11] for the uniform model, in [13] to the biased model and in [6] to the Markov model. Nevertheless, the results are expressed through a generating function, which yields two problems: the *computational complexity* and the *numerical stability*. Clearly, when the size of the set $\mathcal{H}$ or/and the sequence become large, a naive computation for a given n – such as the computation by induction [7] – is computationally expensive. The computation also turns out to be quickly untractable with the improved implementation based on the symbolic system *Combstruct*. Moreover, numerical instability appears that can only be avoided with a careful and tricky implementation [14]. The same problems arise with the software *RegExpCount*. The set $\mathcal{H}$ is viewed as a regular expression, the associated automaton is built and the generating function follows [15]. Our first result in this section is a further writing of the generating function in a more explicit form.

Theorem 2. *Let $\mathcal{H}$ be a set of q words and $F_n(\mathcal{H})$ be the probability that at least one word in $\mathcal{H}$ occurs in a random sequence of size n. The generating function $F_{\mathcal{H}}(z) = \sum_{n\geq 0} F_n(\mathcal{H}) z^n$ satisfies*

$$F_{\mathcal{H}}(z) = \frac{1}{1-z} - \frac{1}{1-z+Q_{\mathcal{H}}(z)}. \tag{2}$$

Proof. Our proof relies on new expressions for matrices $\mathbb{D}(z)$ and $\mathbb{M}(z)$ in the Bernoulli model that extend to the Markov model.

Proposition 1. *The inverse matrix of the fundamental counting matrix satisfies*

$$\mathbb{D}(z)^{-1} = \frac{\mathbb{A}^{-1}(z)}{1-z}\left(\mathbb{I} - \frac{\mathbb{H}(z)\mathbb{A}^{-1}(z)}{1-z+\mathrm{Trace}(\mathbb{H}(z)\mathbb{A}^{-1}(z))}\right). \tag{3}$$

The minimal matrix $\mathbb{M}(z)$ *satisfies*

$$\mathbb{M}(z) = \mathbb{I} - \mathbb{A}^{-1}(z)\left(\mathbb{I} - \frac{\mathbb{H}(z)\mathbb{A}^{-1}(z)}{1-z+\mathrm{Trace}(\mathbb{H}(z)\mathbb{A}^{-1}(z))}\right).$$

Proof. Let us call a 1-*matrix* any matrix whose rows are all equal. One has that any 1-matrix $\mathbb{B}$ satisfies $(\mathbb{I}+\mathbb{B})^{-1} = \mathbb{I} - (1+\mathrm{Trace}(\mathbb{B}))^{-1}\mathbb{B}$. The main arguments for the proof of Prop. 1 are that $\mathbb{A}(z)$ can be inverted in some disc around 0 and that, for any integer $i \geq 0$, $\mathbb{H}(z)\mathbb{A}(z)^{-i}$ is a 1-matrix (details are omitted).

Let us return to the proof of Theorem 2. Let $R_q(\mathcal{H})$ be the probability that the first occurrence of a word from set $\mathcal{H}$ ends at position q. It follows from the definition that $F_n(\mathcal{H}) = \sum_{q\leq n} R_q(\mathcal{H})$. Hence, $F_{\mathcal{H}}(z) = \frac{1}{1-z}\sum_n R_n(\mathcal{H})z^n = \frac{R(z)}{1-z}$. Using Eq. (3), we rewrite $\mathbb{H}\mathbb{D}(z)^{-1} = \frac{\mathbb{B}}{1-z}[\mathbb{I} - \frac{\mathbb{B}}{1-z+\mathrm{Trace}(\mathbb{B})}]$ with $\mathbb{B} = \mathbb{H}(z)\mathbb{A}^{-1}(z)$. As $\mathbb{B}^2 = \mathrm{Trace}(\mathbb{B})\mathbb{B}$ and $\mathrm{Trace}(\mathbb{B}) = Q_{\mathcal{H}}(z)$, we get $\mathbb{H}\mathbb{D}(z)^{-1} = \frac{\mathbb{B}}{1-z+Q_{\mathcal{H}}(z)}$. One rewrites $H(z) = (\,1\,0\,\cdots\,0\,)\mathbb{H}(z)$. Then, Eq. (1) yields $R(z) = (\,1\,0\,\cdots\,0\,)\mathbb{H}(z)\mathbb{D}(z)^{-1}\mathbf{1}_q{}^{\mathrm{t}}$, and finally $R(z) = \frac{1}{1-z+Q_{\mathcal{H}}(z)}(\,1\,0\,\cdots\,0\,)\mathbb{B}\mathbf{1}_q{}^{\mathrm{t}}$. For any matrix $\mathbb{M}$, the product $(\,1\,0\,\cdots\,0\,)\mathbb{M}\mathbf{1}_q{}^{\mathrm{t}}$ is the sum of all coefficients in the first row. For a 1-matrix, this sum is equal to the trace. It follows that $R(z) = \frac{Q_{\mathcal{H}}(z)}{1-z+Q_{\mathcal{H}}(z)}$ and decomposition $\frac{Q_{\mathcal{H}}(z)}{(1-z)(1-z+Q_{\mathcal{H}}(z))} = \frac{1}{1-z} - \frac{1}{1-z+Q_{\mathcal{H}}(z)}$ yields Eq. (2).

Our second result deals with the *co-occurrences* of two signals. Given two sets $\mathcal{H}_1$ and $\mathcal{H}_2$, one studies the probability to have one occurrence from each set in a given short sequence. Typically, each set represents potential binding sites for a regulatory protein.

Theorem 3. *Let* $\mathcal{H}_1$ *and* $\mathcal{H}_2$ *be two disjoint sets of words of size* m*. Denote* $\mathcal{H} = \mathcal{H}_1 \cup \mathcal{H}_2$*. Let* $F_n^{[1,1]}(\mathcal{H}_1, \mathcal{H}_2)$ *be the probability that at least one word in* $\mathcal{H}_1$ *and one word in* $\mathcal{H}_2$ *occur in a random sequence of size* n*. The generating function* $F^{[1,1]}_{\mathcal{H}_1,\mathcal{H}_2}(z) = \sum_n F_n^{[1,1]}(\mathcal{H}_1, \mathcal{H}_2)z^n$ *satisfies*

$$F^{[1,1]}_{\mathcal{H}_1,\mathcal{H}_2}(z) = \frac{1}{1-z} - \frac{1}{1-z+Q_{\mathcal{H}_1}(z)} - \frac{1}{1-z+Q_{\mathcal{H}_2}(z)} + \frac{1}{1-z+Q_{\mathcal{H}}(z)}, \tag{4}$$

where $Q_{\mathcal{H}}(z), Q_{\mathcal{H}_1}(z)$ *and* $Q_{\mathcal{H}_2}(z)$ *are the fundamental multioccurrence series of* $\mathcal{H}$*,* $\mathcal{H}_1$ *and* $\mathcal{H}_2$ *respectively.*

Proof. Let us consider a text of length n with n_1 $\mathcal{H}_1$-occurrences and n_2 $\mathcal{H}_2$-occurrences. The total number of $\mathcal{H}$-occurrences is $n_1 + n_2$ and we have

$$P((n_1 > 0) \cap (n_2 > 0)) = 1 - P(n_1 = 0) - P(n_2 = 0) + P(n_1 + n_2 = 0).$$

Since by Theorem 2, $[z^n]\frac{1}{1-z+Q_{\mathcal{H}}(z)}$ is the probability that there is no word from $\mathcal{H}$ in a text of size n, this equality translates to Eq. (4).

Remark. These results generalize to a Markov process. As for the mean and variance [5], the two contributions due to the overlapping structure of $\mathcal{H}$ and the Markovian process are almost independent. Indeed, it turns out that the results are some functions of the *sum* of $\mathbb{A}(z)$ and $\mathbb{N}(z)$, that represent the dependency to the overlapping structure and the dependency to the Markov process characteristics, respectively. An induction approach [16, 17] does not achieve such a separation. A complexity improvement follows.

3.2 Practical Computation

Our practical results rely on simple observations. When the size of the text increases, the probability to find at least one occurrence increases from 0 to 1. Indeed, for small (respectively large) n, $F_n(\mathcal{H})$ is exponentially close to 0 (respectively 1). Hence, the range where $F_n(\mathcal{H})$ is a meaningful criteria and worth study is in between. The location and size of this "twilight zone" depend on the expected value of $O_n(\mathcal{H})$, e.g. $P(\mathcal{H})$, *and* the number of sequences where the set $\mathcal{H}$ is searched for. Therefore, we assume below that $nP(\mathcal{H})$ is smaller than 1. In this range, an asymptotic expansion turns out to be very tight. More details on the relationship between this bound and the number of sequences will be given in an extended paper.

Theorem 4. *When $nP(\mathcal{H})$ is upper bounded by 1, one has*

$$F_n(\mathcal{H}) = 1 - \left(1 + P(\mathcal{H}) - \widetilde{C}(\mathcal{H})\right)^{-n} \left(1 + O\left(\tfrac{1}{n}\right)\right). \tag{5}$$

The co-occurrence probability satisfies

$$F_n^{[1,1]}(\mathcal{H}_1, \mathcal{H}_2) \sim 1 + e^{-n\log(1+P(\mathcal{H})-C(\mathcal{H}))} - e^{-n\log(1+P(\mathcal{H}_1)-C(\mathcal{H}_1))} - e^{-n\log(1+P(\mathcal{H}_2)-C(\mathcal{H}_2))}.$$

Remarks. Numerical evaluation shows that these formulae are very tight. Due to the simplicity of its computation, our formulae favorably compare to intricate and unstable computation by induction. Expansion (5) also is an improvement on a common numerical approximation [18, 19]

$$F_n(\mathcal{H}) = 1 - (1 - P(\mathrm{H}))^{n-m+1}.$$

Unfortunately, this approximation does not take the words overlaps into account, although they do induce a significant change [20]. Moreover, it is not valid in the finite range. When $\mathcal{H}$ reduces to a single pattern H, a tighter approximation holds in both cases. Namely $\frac{P(\mathrm{H})}{A_{\mathrm{H}}(1)}$ is slightly more accurate than $P(\mathcal{H}) - \widetilde{C}(\mathcal{H})$.

Proof. Generating functions in Eq. (2) and Eq. (4) depend on the *fundamental multioccurrence series* properties. Lemma below provides a uniform approximation for the fundamental multioccurrence series.

Lemma 1. *The fundamental multioccurrence series of a set $\mathcal{H}$ satisfies*

$$Q_{\mathcal{H}}(z) = P(\mathcal{H})z^m - \sum_{\mathrm{H},\mathrm{F}\in\mathcal{H}} \sum_{w\in\widetilde{\mathcal{C}}_{\mathrm{H},\mathrm{F}}} P(\mathrm{H}w)z^{|\mathrm{H}w|} + O\left(m\,P(\mathcal{H})^2\right). \quad (6)$$

The root of smallest modulus of $1 - z + Q_{\mathcal{H}}(z)$ is a real positive number ρ that is greater than 1 and satisfies

$$\rho - (1 + P(\mathcal{H}) - \widetilde{C}(\mathcal{H})) = O(mP(\mathcal{H})^2) \ . \quad (7)$$

Proof. We first prove that

$$\sum_{j=1}^{m}[z^{m+j}]\,\mathrm{Trace}(\mathbb{H}\mathbb{A}^{-1}(z)) = \sum_{j=1}^{m}[z^{m+j}]Q_{\mathcal{H}}(z) = \sum_{\substack{\mathrm{H},\mathrm{F}\in\mathcal{H}\\ w\in\widetilde{\mathcal{C}}_{\mathrm{H},\mathrm{F}}}} P(\mathrm{H}w)z^{|\mathrm{H}w|}. \quad (8)$$

One has $\mathbb{A}^{-1}(z) = \mathbb{I} + \sum_{k\geq 1}(-1)^k\mathbb{C}^k$. A non-zero term in $[z^\ell]\mathbb{H}\mathbb{C}^k$ is mapped to a word of length ℓ which is an overlapping chain of words from $\mathcal{H}$. The weight of each chain is its probability. Such a word w can be decomposed unambiguously as a product $\mathrm{H}\cdot\widetilde{c}_1\cdots\widetilde{c}_j$ of one word $\mathrm{H}\in\mathcal{H}$ and a product of j *minimal right complements*. Assume now that $m < |w| < 2m$. There are $\binom{j-1}{r-1}$ choices of grouping consecutive words among the $\widetilde{c}_i$'s in order to get a valid decomposition $w = \mathrm{H}c_1\cdots c_r$ where the c_i's are right complements. Therefore, the contribution of w to $\mathrm{Trace}(\mathbb{H}(\mathbb{I}+\sum_{k\geq 1}(-1)^k\mathbb{C}^k))$ is $z^\ell\,P(w)\sum_{r=1}^{j-1}\binom{j-1}{r-1}(-1)^r$. This sum is $z^\ell\,P(w)$ if $j = 1$ (which means $w = \mathrm{H}\widetilde{c}$ with $\widetilde{c}$ a minimal right complement) and 0 otherwise. Eq. (8) is established.

A simple combinatorial argument provides for $\sum_{\ell\geq 2m}[z^\ell]\,\mathrm{Trace}(\mathbb{H}\mathbb{A}^{-1}(z))$ the upper bound $O(mP(\mathcal{H})^2)$. Indeed, each monomial in the sum is associated to an overlapping chain w of $\mathcal{H}$-words. Chain w rewrites unambiguously $\mathrm{H}_1 w_1 \mathrm{H}_2 w_2$ where H_1 its prefix in $\mathcal{H}$ and H_2 the first $\mathcal{H}$-word that goes beyond position $2m$. The overall probability of such events is trivially upper bounded by $P(\mathcal{H})mP(\mathcal{H})$.

We study now the zeros of the equation $g(z) = 1 - z + Q_{\mathcal{H}}(z)$. For small values of $P(\mathcal{H})$, a *bootstrapping* approach [21, 22] allows for a derivation of the local development of ρ given in Eq. (7).

The Darboux theorem for the series $\phi(z)/g(z)$ implies that the n-th coefficient of this series satisfies $p_n \sim \frac{\phi(\rho)}{\rho g'(\rho)}\rho^{-n}$. General and detailed results on its use on rational series can be found in [23]. Using Eq. (7) for sets $\mathcal{H}$, $\mathcal{H}_1$ and $\mathcal{H}_2$ yields Theorem 4. As $P(\mathcal{H}) = O(\frac{1}{n})$ in this range, we get the approximation order.

4 Efficient Computation of Fundamental Parameters

To put formulae into effect, one needs to compute efficiently for a set of words $\mathcal{H}$ the fundamental quantities of Definition 4. First we present a general method for an arbitrary set $\mathcal{H}$ based on a classical algorithm. If $\mathcal{H}$ has a particular structure (i.e. consists of approximate words), a more efficient way is available.

4.1 Correlation for an Arbitrary Set of Words

We resort in this section to the well-known Aho-Corasick algorithm [24, 25] which builds from a finite set of words $\mathcal{H}$ a deterministic complete automaton (not necessarily minimal) recognizing the language $\Sigma^*\mathcal{H}$ where Σ is the (finite) alphabet. This automaton is the basis of many efficient algorithms on string matching problems and is often called the *string matching automaton*. We use a variant represented as a trie together with a failure function. Let $\mathcal{T}_\mathcal{H}$ be the ordinary trie representing $\mathcal{H}$, seen as a finite deterministic automaton $(Q, \delta, \varepsilon, T)$ where the set of states is $Q = \mathit{Pref}(\mathcal{H})$ (prefixes of words in $\mathcal{H}$), the initial state is ε, the set of final states is $T = \Sigma^*\mathcal{H} \cap \mathit{Pref}(\mathcal{H})$ and the incomplete transition function δ is defined by $\delta(p, a) = pa$ if $pa \in \mathit{Pref}(\mathcal{H})$ and undefined otherwise. For a word $u \in \mathit{Pref}(\mathcal{H})$, the failure function *Border* associates

$$\mathit{Border}(u) = \text{the longest proper suffix of } u \text{ which belongs to } \mathit{Pref}(\mathcal{H}).$$

In the following we identify a word $u \in \mathit{Pref}(\mathcal{H})$ with the node at the end of the branch labelled by u, so that *Border* defines also a map on the nodes of the tree. There are efficient $O(|\mathcal{H}|)$ algorithms [24, 25] linear both in time and space building such a tree structure together with the auxiliary *Border* function.

For any set $\mathcal{H}$ and $w \in \mathcal{H}$, one can compute $C_{w,\mathcal{H}}$ and $\widetilde{C}_{w,\mathcal{H}}$ using this structure. Indeed, we associate to $w \in \mathcal{H}$ a suffix chain of nonempty words $(u_1, \ldots, u_k)$ obtained by successive application of the failure function before getting ε. Then the trie $\mathcal{T}$ of the right complements of H in $\mathcal{H}$ is obtained by merging all the subtries rooted at the nodes labelled by the u_i's. The trie $\widetilde{\mathcal{T}}$ of the minimal complements of H in $\mathcal{H}$ corresponds to the pruning of $\mathcal{T}$ where, along any branch, we only keep the nodes from the root to the first terminal node (see Fig. 1). One can easily compute the quantities $C(\mathcal{H})$ and $\widetilde{C}(\mathcal{H})$ (or $P(\mathcal{H})$) along the construction of these trees.

4.2 Approximate Words and Generalized Consensus

In this section, we present algorithms to compute the occurrence probability and the correlation factor (but due to limitations of our approach not the minimal correlation factor) for a certain kind of set of words.

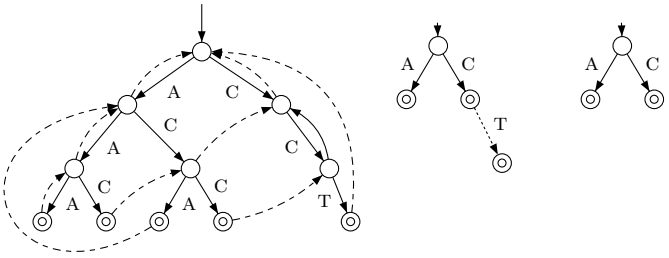

Fig. 1. For the set $\mathcal{H} = \{\text{AAA}, \text{AAC}, \text{ACA}, \text{ACC}, \text{CCT}\}$: The trie $\mathcal{T}_\mathcal{H}$ with (dashed) links *Border* (left), the trie of the right complements of AAC in $\mathcal{H}$ (middle) and the trie of the minimal right complements of AAC in $\mathcal{H}$.

Definition 5. *A positional pattern of length m is a sequence of sets of letters $(\mathcal{H}_i)_{i=1}^m$. Each set $\mathcal{H}_i \subseteq \Sigma$ is the set of symbols permitted at position i among symbols of the alphabet Σ.*

Hereafter we identify a positional pattern and the set of words it represents. For instance, the pattern $\mathcal{H} = (\{\text{A}, \text{C}\}, \{\text{T}\}, \{\text{A}, \text{T}\})$ is the set $\{\text{ATA}, \text{ATT}, \text{CTA}, \text{CTT}\}$. To define a neighborhood, we will use in this section the *Hamming distance* d.

Definition 6. *A* neighborhood *of $\mathcal{H} = (\mathcal{H}_i)_{i=1}^m$ is a set of words $\mathcal{N} = (\mathcal{N}_i)_{i=1}^m$ such that $\mathcal{H} \subseteq \mathcal{N}$ (i.e. $\mathcal{H}_i \subseteq \mathcal{N}_i$ for $i = 1..m$). The k-neighborhood (relative to $\mathcal{N}$) $B_k(\mathcal{H})$ of center $\mathcal{H}$ and radius k is $B_k(\mathcal{H}) = \{w \in \mathcal{N} \mid d(\mathcal{H}, w) \leq k\}$.*

The most natural example consists in a center $\mathcal{H}$ reduced to one word of length m and a neighborhood $\mathcal{N} = \Sigma^m$ where all letters are allowed. However our model enables us to consider words with some forbidden errors at specific positions.

The set of words $\mathcal{H}$ is called below the *center* of the neighbourhood. It is the so-called *consensus* of the biologists. The motifs in the neighbourhood are the *approximate words*, also called *spurious motifs* [18]. They appear in the extraction of regulatory signals. Our definition – and our algorithms as well – covers the case where the substitutions occur at some specific positions. It also covers the case of an extended alphabet, for instance the fifteen IUPAC ambiguity code [26]. Symmetric structures can also be addressed, as in the special case of palindromes, where the errors must maintain the palindromic structure. This case is of interest for the study of restriction-modification systems [2].

Note that when k errors are allowed, the number of words in the neighborhood is $O(m^k)$. Therefore, a naive algorithm that examines all the words of the neighborhood is an exponential algorithm with respect to k. Our algorithm is polynomial in m and k for both Bernoulli and Markov models. The key idea is to consider a word or a set of words as a formal series. One knows [23] that given a rational language, substituting probabilities to symbols yields in the Bernoulli model the probability occurrence. Finally if we know in advance the number of errors allowed, we can use truncated expansion of the series.

Occurrence probability for approximate words. In this section, we compute the occurrence probability of the neighborhood $B_k(\mathcal{H})$.

Definition 7. *For a pattern $\mathcal{H}$ of length m and a neighborhood $\mathcal{N}$, let us define the generating function*

$$F_{\mathcal{H}}(u) = \sum_{w \in \mathcal{N}} P(w) u^{d(\mathcal{H}, w)}. \tag{9}$$

The occurrence probability of $B_k(\mathcal{H})$ is $P(B_k(\mathcal{H})) = \sum_{i=0}^k [u^i] F_{\mathcal{H}}(u)$.

Bernoulli model. In this model, Eq. (9) rewrites $F_{\mathcal{H}}(u) = \prod_{i=1}^m (P(\mathcal{H}_i) + u\, P(\mathcal{N}_i \setminus \mathcal{H}_i))$. Since we are only interested in coefficient degree less than k, it is enough to compute the truncated expansion up to degree k. Remark that k is known in advance and usually small in applications.

ProbabilityBernoulli$(k, \mathcal{H}, \mathcal{N})$
1 $f(u) \leftarrow 1$
2 **for** $i \leftarrow 1$ **to** m **do**
3 $\quad f(u) \leftarrow f(u) \times (P(\mathcal{H}_i) + u\, P(\mathcal{N}_i \setminus \mathcal{H}_i))$
4 **return** $\sum_{i=0}^{k} [u^i] f(u)$

Proposition 2. *The algorithm* ProbabilityBernoulli*(), for a Bernoulli model, computes the probability $P(B_k(\mathcal{H}))$ with $O(mk)$ time complexity and $O(k)$ space complexity.*

One needs only to store one polynomial $f(u)$ of degree k at a time. Moreover the iterative step 3, updating $f(u)$, can be computed *in place* so no extra storage is needed and requires only $O(k)$ operations since we need to multiply a polynomial of degree k with a polynomial of degree at most 1.

Markov model. For a Markov model of order $p > 0$, the stationary probability of w is $P(w) = \sum_{c \in \Sigma^p} \pi_c\, P(w|c)$, with $P(w|c)$ the probability that H occurs after c, and π_c the stationary probability of c. So in this context, Eq. (9) rewrites

$$F_{\mathcal{H}}(u) = \sum_{c \in \Sigma^p} \pi_c f_{\mathcal{H},c}(u), \quad \text{where } f_{\mathcal{H},c}(u) = \sum_{w \,\in\, \mathcal{N}} P(w|c) u^{d(w,\mathcal{H})}.$$

In the following algorithm we will need two notations. For a set S, $\delta_S(j) = 1$ if $j \in S$ and 0 otherwise. For a word $c \in \Sigma^p$ and a symbol $j \in \Sigma$, the *j-shift* $\sigma_j(c)$ of the word c by j is the word of length p obtained by erasing the first letter of c and adding the symbol j at the end. For instance, $\sigma_{\mathrm{G}}(\mathrm{ATC}) = \mathrm{TCG}$.

ProbabilityMarkov$(k, \mathcal{H}, \mathcal{N})$
1 **for** $c \in \Sigma^p$ **do**
2 $\quad f_c(u) \leftarrow 1$
3 **for** $i \leftarrow m$ **downto** 1 **do**
4 $\quad$ **for** $c \in \Sigma^p$ **do**
5 $\quad\quad f'_c(u) \leftarrow 0$
6 $\quad\quad$ **for** $j \in \Sigma$ **do**
7 $\quad\quad\quad c' \leftarrow \sigma_j(c)$
8 $\quad\quad\quad f'_c(u) \leftarrow f_{c'}(u) \times P(j|c) \times \delta_{\mathcal{N}_r}(j) u^{1-\delta_{\mathcal{H}_i}(j)}$
9 $\quad$ **for** $c \in \Sigma^p$ **do**
10 $\quad\quad f_c(u) \leftarrow f'_c(u)$
11 ▷ Use the stationary distribution π to obtain the result
12 $F(u) \leftarrow 0$
13 **for** $c \in \Sigma^p$ **do**
14 $\quad F(u) \leftarrow F(u) + \pi_c \times f_c(u)$
15 **return** $\sum_{i=0}^{k} [u^i] F(u)$

Without proof, let us state the complexity of this algorithm.

Proposition 3. *The algorithm* ProbabilityMarkov*() computes for a Markov model of order p, the probability $P(B_k(\mathcal{H}))$ with $O(mV^{p+1}k)$ time complexity and $O(V^p k)$ space complexity where V is the cardinal of the alphabet, k is the number of errors and $\mathcal{H}$ is of length m.*

Overlap factor for approximate words. Our aim here is to provide a symbolic computation of the *overlap factor* $C(B_k(\mathcal{H}))$. Similarly to Section 4.2, for each consecutive possible overlap position i, we consider a bivariate polynomial with the Hamming distance d where u marks the number of errors relatively to a prefix of length m and v marks the numbers of errors relatively to a suffix of size m

$$D_{i,\mathcal{H}}(u,v) = \sum_{\substack{\mathrm{F},\mathrm{G}\in\mathcal{N} \\ \text{overlapping at } i}} P(\mathrm{F}\,\mathrm{G}^m_{m-i+1}) u^{d(\mathcal{H},\mathrm{F})} v^{d(\mathcal{H},\mathrm{G})}.$$

The following algorithm computes $C(B_k(\mathcal{H}))$ and has $O(m^2k^2)$ time complexity and $O(k^2)$ space complexity. Note also that this algorithm can be readily extended as in Section 4.2 to a Markov model of order p with alphabet of cardinality V with a time complexity $O(m^2V^{p+1}k^2)$.

OverlapFactorBernoulli$(k, \mathcal{H}, \mathcal{N})$

```
1  D(u,v) ← 0
2  for i ← 2 to m − 1 do
3        f(u,v) ← 1
4        for j ← m + i − 1 to m do
5              f(u,v) ← f(u,v) × ( P(H_j) + v P(N_j \ H_j) )
6        for j ← m to i do
7              f(u,v) ← f(u,v) × ( P(H_j ∩ H_{j−i+1})
                          +u P((N_j \ H_j) ∩ H_{j−i+1}) + v P((N_{j−i+1} \ H_{j−i+1}) ∩ H_j)
                          +uv P((N_j \ H_j) ∩ (N_{j−i+1} \ H_{j−i+1})) )
8        for j ← i − 1 to 1 do
9              f(u,v) ← f(u,v) × ( P(H_j) + u P(N_j \ H_j) )
10       D(u,v) ← d(u,v) + f(u,v)
11 return Σ_{0≤i,j≤k} [u^i v^j] D(u,v)
```

5 Conclusion

We provided efficient algorithms to assess the significance of exceptional words in a long sequence – typically a whole genome – or a set of small sequences. While the approaches to word counting through recurrences suffer from a combinatorial explosion when the text is Markovian or when the signal is strongly degenerated, our formulae or algorithms allow for a fast (polynomial) computation. We are currently working on a possible extension to structured motifs or dyads [8]. Among the possible applications, it might be interesting to compile regulatory motifs in eucaryotic genomes, possibly the human genome, and to evaluate their significance.

References

1. Panina, E., Mironov, A., Gelfand, M.: Statistical analysis of complete bacterial genomes:Avoidance of palindromes and restriction-modification systems. Mol. Biol. **34** (2000) 215–221

2. Vandenbogaert, M., Makeev, V.: Analysis of bacterial RM-systems through genome-scale analysis and related taxonomic issues. In Silico Biol. **3** (2003) 12
3. Robin, S., Schbath, S.: Numerical comparison of several approximations on the word count distribution in random sequences. J. Comput. Biol. **8** (2001) 349–359
4. Chiang, D., Moses, A., Kellis, M., Lander, E., Eisen, M.: Phylogenetically and spatially conserved word pairs associated with gene-expression in yeasts. Genome Biol. **4:R43** (2003)
5. Régnier, M., Szpankowski, W.: On pattern frequency occurrences in a Markovian sequence. Algorithmica **22** (1997) 631–649
6. Régnier, M.: A unified approach to word occurrences probabilities. Discrete Appl. Math. **104** (2000) 259–280 Special issue on Computational Biology.
7. Robin, S., Daudin, J.J.: Exact distribution of word occurrences in a random sequence of letters. J. Appl. Prob. **36** (1999) 179–193
8. Robin, S., Daudin, J.J., Richard, H., Sagot, M., Schbath, S.: Occurrence probability of structured motifs in random sequences. J. Comput. Biol. **9** (2001) 761–773
9. Pevzner, P., Borodovski, M., Mironov, A.: Linguistics of nucleotide sequences i: the significance of deviations from mean statistical characteristics and prediction of the frequencies of occurrence of words. J. Biomol. Struct. Dynam. **6** (1989) 1013–1026
10. Bender, E.A., Kochman, F.: The Distribution of Subwords Counts is Usually Normal. European J. Combin. **14** (1993) 265–275
11. Guibas, L., Odlyzko, A.: String Overlaps, Pattern Matching and Nontransitive Games. J. Combin. Theory Ser. A **30** (1981) 183–208
12. Tanushev, M., Arratia, R.: Central limit theorem for renewal theory for several patterns. J. Comput. Biol. **4** (1997) 35–44
13. Régnier, M., Szpankowski, W.: On the approximate pattern occurrences in a text. In: Compression and Complexity of SEQUENCES, IEEE Computer Society (1997) 253–264
14. Klaerr-Blanchard, M., Chiapello, H., Coward, E.: Detecting localized repeats in genomic sequences: A new strategy and its application to *B. subtilis* and *A. thaliana* sequences. Comput. Chem. **24** (2000) 57–70
15. Nicodème, P., Salvy, B., Flajolet, P.: Motif statistics. Theoret. Comput. Sci **287** (2002) 593–618
16. Chrysaphinou, C., Papastavridis, S.: The occurrence of sequence of patterns in repeated dependent experiments. Theory Probab. App. **79** (1990) 167–173
17. Szpankowski, W.: Average Case Analysis of Algorithms on Sequences. John Wiley and Sons, New York (2001)
18. Buhler, J., Tompa, M.: Finding Motifs Using Random Projections. In: RECOMB'01, ACM (2001) 69–76
19. Beaudoing, E., Freier, S., Wyatt, J., Claverie, J., Gautheret, D.: Patterns of Variant Polyadenylation Signal Usage in Human Genes. Genome Res. **10** (2000) 1001–1010
20. van Helden, J., André, B., Collado-Vides, J.: Extracting regulatory sites from the upstream region of yeast genes by computational analysis of oligonucleotide frequencies. J. Mol. Biol. **281** (1998) 827–842 http://rsat.ulb.ac.be/rsat/.
21. Knuth, D.: The average time for carry propagation. Indag. Math. **40** (1978) 238–242
22. Régnier, M.: Mathematical tools for regulatory signals extraction. In Kolchanov, N., Hofestaedt, R., eds.: Bioinformatics of Genome Regulation and Structure, Kluwer Academic Publisher (2004) 61–70
23. Flajolet, P., Sedgewick, R.: Analysis of Algorithms. Addison-Wesley (1996)

24. Aho, A.V., Corasick, M.J.: Efficient string matching: an aid to bibliographic search. Commun. ACM **18** (1975) 333–340
25. Crochemore, M., Rytter, W.: Jewels of Stringology. World Scientific Publishing, Hong-Kong (2002) 310 pages.
26. Blanchette, M., Sinha, S.: Separating real motifs from their artifacts. Bioinformatics (ISMB special issue) **817** (2001) 30–38