Constructions for Clumps Statistics.

Frédérique Bassino, Julien Clément, Julien Fayolle, Pierre Nicodème

Constructions for Clumps Statistics

F. Bassino[1], J. Clément[2], J. Fayolle[3], and P. Nicodème[4]

[1]*LIPN, CNRS-UMR 7030, Université Paris-Nord, 93430 Villetaneuse, France.* bassino@lipn.univ-paris13.fr
[2]*GREYC, CNRS-UMR 6072, Université de Caen, ENSICAEN, 14032 Caen, France.* clement@info.unicaen.fr
[3]*LRI, CNRS-UMR 8623, Univ. Paris-Sud, Bât 490, 91405 Orsay, France.* Julien.Fayolle@lri.fr
[4]*LIX, CNRS-UMR 7161, École polytechnique, 91128 Palaiseau, France.* nicodeme@lix.polytechnique.fr

We consider a component of the word statistics known as clump; starting from a finite set of words, clumps are maximal overlapping sets of these occurrences. This object has first been studied by Schbath [22] with the aim of counting the number of occurrences of words in random texts. Later work with similar probabilistic approach used the Chen-Stein approximation for a compound Poisson distribution, where the number of clumps follows a law close to Poisson. Presently there is no combinatorial counterpart to this approach, and we fill the gap here. We also provide a construction for the yet unsolved problem of clumps of an arbitrary finite set of words. In contrast with the probabilistic approach which only provides asymptotic results, the combinatorial method provides exact results that are useful when considering short sequences.

Keywords: Words counting, formal language decomposition, generating functions, automata

1 Introduction

Counting words and motifs in random texts has provided extended studies with theoretical and practical reasons. Much of the present combinatorial research has built over the work of Guibas and Odlyzko [9, 10] who defined the autocorrelation polynomial of a word. As an apparently surprising consequence of their work, the mean waiting time for the first occurrence (or expectation of the number of characters read until finding the first occurrence) of the word 111 in a Bernoulli string with probability $1/2$ for zeroes and ones is larger than the mean waiting time for the first occurrence of the word 100. This is due to the fact that the words 111 occur by *clumps* or sets of overlapping occurrences of 111 or, equivalently, inside runs of at least three ones, the probability of extending a clump or a run by one position being $1/2$; this implies that the average number of 111 in a clump is larger than one; in contrast, there is only one 100 in each clump of 100. Since the probability that the word 111 and the word 100 start at a given position both are $1/8$, the expectation of the interarrival time (or of the number of characters needed to find a new occurrence once a clump has been read) of clumps of 111 is larger than the expectation of the interarrival time of clumps of 100.

Historically, multiple word counting used different approaches for analysis of the reduced (and easier) case where no word can be a factor of another word of the considered set of words and the general (and harder) case. We analyze first in this article several statistics connected to clumps of a pattern, where the pattern is one word or a reduced set of words. Our approach is based on properties of the Régnier-Szpankowski [18] decomposition of languages along occurrences of the considered word or set of words

and on properties of the prefix codes generating the clumps. We provide explicit generating functions in the Bernoulli model for statistics such as (i) the number of clumps, (ii) the number of occurrences of words of the pattern, (iii) the number of k-clumps (clumps with exactly k occurrences of the words of the pattern), (iv) the number of positions of the texts covered by clumps; these explicit results may be extended to a Markov model, providing some technicalities. We get also to the same results in the Bernoulli model by an algorithmic approach where we construct deterministic finite automata recognizing clumps in the general case. This approach extends directly to the Markov model. We obtain as a direct consequence a Gaussian limit law for the number of clumps and the size of texts covered by clumps in random texts in the Bernoulli and Markov model.

Consider a rough first approximation for clumps of one word. If the probability occurrence of a word w is small, the number of clumps of this word is likely to be also small. Then the number of clumps in texts of size n is close to a Poisson law of parameter $\lambda = n \times \mathbf{P}(\text{a clump starts at position } i)$, where i is a random position. Approximating further, the random number of occurrences of the word w in a clump follows a geometric law with parameter ω, where ω is the probability of self-overlap of the word. Schbath [22] gave the first moment of the number of k-clumps (clumps where w occurs exactly k times) and of the number of clumps of one word in Bernoulli texts. Reinert and Schbath [19] obtained in the Markov case of any order a compound Poisson limit law for the count of number of occurrences of reduced sets of words by the Chein-Stein method. See Reinert *et al.* [20] and Pape [16] for a review of this approach and Barbour *et al.* [1] for an extensive introduction to the Poisson approximation. Recently, Stefanov *et al.* [24] used a stopping time method to compute the distribution of clumps; their results are not explicit and practical application of their method requires the inversion of a probability generating function. Roquain and Schbath [21] provide also a probabilistic approach to the number of clumps. Both approach are limited to the reduced case. In the context of analysing the insertion depth in suffix-trees, Jacquet and Szpankowski [13] computed by combinatorial methods the generating function of clumps of suffixes of a word.

We describe in Section 2 our notations. Section 3 describes the formal language approach based on previous work of Régnier and Szpankowski and Section 4 provides the automaton construction for the general case. We prove limit laws for the number of clumps and the size of the texts covered by clumps in Section 5.

2 Preliminaries

We consider a finite alphabet $\mathcal{A}$. Unless explicitly stated when considering a Markov source, the texts are generated by a non-uniform Bernoulli source over the alphabet $\mathcal{A}$. Given a set of words, clumps of these words may be seen as a generalization of runs of one letter.

Reduced set of words. A set of words $\mathcal{V} = \{v_1, \ldots, v_r\}$ is reduced if no v_i is factor of a v_j with i different of j. For instance the set $\{aa, aba\}$ is reduced whereas the sets $\{aa, aab\}$, $\{aa, baa\}$, $\{aa, baab\}$ are non-reduced.

Clumps and k-clumps. When considering a set of words $\mathcal{V} = \{v_1, \ldots, v_r\}$ where each word v_i has size at least 2, a clump is a maximal set of occurrences of words of $\mathcal{V}$ such that

– any two consecutive letters of the clump belong to (is a factor of) at least one occurrence of a word from $\mathcal{V}$,

– either the clump is composed of a single occurrence that overlaps no other occurrence, or each occurrence *overlaps* at least one other occurrence.

This definition naturally applies also to the case where $\mathcal{V}$ is composed of a single word.

Goulden and Jackson [8] considered *clusters* of words. We use as intermediate step the related notion of *clustering-word*.

Definition 1 (Clustering words & Clumps) *A* clustering-word *for the set $\mathcal{V} = \{v_1, \ldots, v_r\}$ is a word $w = w[1..|w|] \in \mathcal{A}^*$ such that any two consecutive positions in w are* covered *by the same occurrence in w of a word $v \in \mathcal{V}$. The position i of the word w is* covered *by a word v if $v = w[(j - |v| + 1) \ldots j]$ for some $j \in \{|v|, \ldots, |w|\}$ and $j - |v| + 1 \leq i \leq j$. More formally, w is a clustering-word for the set $\mathcal{V}$ if we have*

$$\forall i \in \{1, \ldots, |w|-1\} \quad \exists\, v \in \mathcal{V}, \exists\, p \in \mathrm{Pos}_w(v) \quad \text{such that} \quad p - |v| + 1 < i + 1 \leq p,$$

where $\mathrm{Pos}_w(v)$ is the set of positions of occurrences of v in w.

A clump *in a text T, generically denoted here by $\mathfrak{K}$, is a maximal clustering-word in the sense that there exists no occurrence of the set $\mathcal{V}$ in T that overlaps this clustering-word without being a factor of it.*

Note that the clusters in the terminology of Goulden and Jackson are exactly subsets of $\mathrm{Pos}_w(v)$ in our settings. As example, considering the set $\mathcal{V} = \{aba, bba\}$ and the text $T = bbbababababababbbbabaababb$, we have

$$T = b\underline{bbababababa}bb\underline{bbbaba}\,\underline{ababa}bb$$

where the clumps are underlined. The word $bbababababa$ beginning at the second position of the text is a clump, and so are the words $bbaba$ and aba beginning at the 15th and 20th positions. On the contrary, the word $ababa$ beginning at the sixth position is not a clump since it is not maximal; neither is a clump the word $bbabaaba$ beginning at the 15th position, since its two-letters factor aa is neither a factor of an occurrence of aba nor of an occurrence of bba.

We remark that a single word is a clustering-word and that, as mentioned previously, a clump may be composed of a single word. A *k-clump* of occurrences of $\mathcal{V}$ (denoted by $\mathfrak{K}_k$) is a clump containing exactly k occurrences of $\mathcal{V}$.

Notations. We consider the *residual* language $\mathcal{D} = \mathcal{L}.w^-$ defined as $\mathcal{D} = \{x,\ x \cdot w \in \mathcal{L}\}$. Considering two languages $\mathcal{L}_1$ and $\mathcal{L}_2$, if we have $\mathcal{L}_1 \subset \mathcal{L}_2$, we write $\mathcal{L}_2 - \mathcal{L}_1 = \mathcal{L}_2 \setminus \mathcal{L}_1$.

Autocorrelations, correlations and right extension sets of words. We recall here the definition of *Right Extension Sets* introduced in Bassino *et al.* [2]. This notion is a generalization of the correlation of two words to the cases where one word may be factor of the other one.

The right extension set of a pair of words (v_1, v_2) is

$$\mathcal{E}_{v_1,v_2} = \{\ e \quad | \quad \text{there exists } e' \in \mathcal{A}^+ \quad \text{such that} \quad v_1 e = e' v_2 \quad \text{with } 0 < |e| < |v_2|\}.$$

The "usual" autocorrelation set $\mathcal{C}_{w,w}$ of the word w (denoted further by $\mathcal{C}$ when there is no ambiguity) is defined as usual by

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

Remark that the empty word ε belongs to the autocorrelation set $\mathcal{C}$. In contrast, we define the *strict* autocorrelation set $\mathcal{C}_\circ$ by $\mathcal{C}_\circ = \mathcal{C} - \{\varepsilon\}$.

We remark also that $\mathcal{C}_\circ$ is empty if the word w has no autocorrelation.

If the word v_1 is not factor of v_2 the right extension set from v_1 to v_2 is the usual correlation set of $\mathcal{C}_{v_1,v_2}$ of v_1 and v_2, defined as,

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

Note also that the correlation set of two words may be empty. When we have $w = v_1 = v_2$, we get $\mathcal{E}_{w,w} = \mathcal{C}_\circ = \mathcal{C} - \varepsilon$.

We have as examples

$$\mathcal{C}_{aabaa,aab} = \{b, ab\}, \qquad \mathcal{C}_{ababa,ababa} = \{\varepsilon, ba, baba\}, \qquad \mathcal{E}_{aaa,aaaa} = \{aa, aaa\}.$$

Number of occurrences of words and clumps. We note respectively O_n^w and $O_n^{\mathfrak{K}}$ the random variables counting the number of occurrences of a word w and the number of clumps of this word in random texts of size n. The random variable $O_n^{\mathfrak{K}_k}$ counts occurrences of k-clumps.

Generating functions. For any language $\mathcal{L}$, we define its generating function in the Bernoulli model by

$$\mathcal{L}(z) = \sum_{w \in \mathcal{L}} \pi_w z^{|w|} = \sum_{n \geq 0} f_n z^n,$$

where π_w is the probability of the word w in the usual Bernoulli model (that is the product of the probability of the letters composing the word, with their multiplicities) and f_n is the probability that a random word of size n belongs to the language $\mathcal{L}$. For instance for a binary alphabet $\{a, b\}$ and a (biased) Bernoulli model with $\pi_a = p$ and $\pi_b = 1 - p$ ($p \in [0, 1]$), the generating function of the language $\mathcal{C}_{ababa,ababa} = \{\varepsilon, ba, baba\}$ is $1 + p(1-p)z^2 + p^2(1-p)^2 z^4$.

We aim here at providing generating functions or explicit formulas for counting the number of clumps, the total size of text covered by clumps or the number of clumps with exactly k occurrences. This typically corresponds when considering the object u to multivariate generating functions such as

$$F_u(z, x) = \sum_{T \in \mathcal{L}} \mathbf{P}(T) z^{|T|} x^{|T|_u} = \sum f_{n,i} x^i z^n \tag{1}$$

where $\mathbf{P}(T)$ is the weight (i.e., probability) of the text T among texts of same size, $|T|_u$ is the number of occurrences of the object u in the text T and $f_{n,i}$ is the probability that a text of size n has i occurrences of this object. This extends naturally for counting more than one object by considering multivariate generating functions with several parameters.

If the random variable X_n counts the number of objects in texts of size n, we get from Equation (1)

$$\mathbf{E}(X_n) = [z^n]\, \frac{\partial F(z,x)}{\partial x}\bigg|_{x=1}, \qquad \mathbf{E}(X_n^2) = [z^n]\, \frac{\partial}{\partial x} x \frac{\partial F(z,x)}{\partial x}\bigg|_{x=1}.$$

Recovering exactly or asymptotically these moments follows then from classical methods.

3 Formal language approach

3.1 Régnier and Szpankowski decomposition

Our work extends the formal language approach of Régnier and Szpankowski [18]. We briefly recall their approach. Considering one word w, Régnier and Szpankowski use a natural parsing or decomposition of texts with at least one occurrence of w, parsing unambiguously a text with at least one occurrence of w as follows:

1. the part of text from the beginning to the first occurrence of the word w belongs to the *Right* language,
2. if there are any other occurrences of the word w, each two consecutive occurrences are separated by a word from the *Minimal* language,
3. the part of text from the last occurrence of w to the end belongs to the *Ultimate* language.

Moreover, there is a *Not* language of texts without any occurrence of the considered word w. Régnier [17] further extended this decomposition approach to a reduced set of words.

We follow here the presentation of Lothaire [14] (Chapter 7). Let $\mathcal{V} = \{v_1, \ldots, v_r\}$ be a reduced set of words. We have, formally

Definition 2 *Right, Minimal, Ultimate and Not languages.*

– *The "Right" language $\mathcal{R}_i$ associated to the word v_i is the set of words*

$$\mathcal{R}_i = \{r \mid r = e \cdot v_i \text{ and there is no } v \in \mathcal{V} \text{ such that } r = xvy \text{ with } |y| > 0\}.$$

– *The "Minimal" language $\mathcal{M}_{ij}$ leading from a word v_i to a word v_j is the set of words*

$$\mathcal{M}_{ij} = \{m \mid v_i \cdot m = e \cdot v_j \text{ and there is no } v \in \mathcal{V} \text{ such that } v_i \cdot m = xvy \text{ with } |x| > 0, |y| > 0\}.$$

– *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*

$$\mathcal{U}_i = \{u \mid \text{ there is no } v \in \mathcal{V} \text{ such that } v_i \cdot u = xvy \text{ with } |x| > 0\}.$$

– *The "Not" language is the set of texts where no word from $\mathcal{V}$ occurs*

$$\mathcal{N} = \{n \mid \text{there is no } v \in \mathcal{V} \text{ such that } n = xvy\}.$$

The notations $\mathcal{R}, \mathcal{M}, \mathcal{U}$ and $\mathcal{N}$ refer here to the Right, Minimal, Ultimate and Not languages of a single word.

We consider as example the word $w = ababa$; in the following texts, the underlined words belong to the set $\mathcal{M}$; the overlined word does not since the occurrence represented in bold faces is an intermediate occurrence.

$$ababa\underline{aaaaababa} \qquad ab\mathbf{ababa}\overline{bbbababa} \qquad ababa\underline{ba}.$$

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

$$\bigcup_{k\geq 1} \left(\mathbb{M}^k\right)_{i,j} = \mathcal{A}^\star \cdot v_j + \mathcal{C}_{ij} - \delta_{ij}\varepsilon, \qquad \mathcal{U}_i \cdot \mathcal{A} = \bigcup_j \mathcal{M}_{ij} + \mathcal{U}_i - \varepsilon, \tag{2}$$

$$\mathcal{A} \cdot \mathcal{R}_j - (\mathcal{R}_j - v_j) = \bigcup_i v_i \mathcal{M}_{ij}, \qquad \mathcal{N} \cdot v_j = \mathcal{R}_j + \bigcup_i \mathcal{R}_i \left(\mathcal{C}_{ij} - \delta_{ij}\varepsilon\right), \tag{3}$$

where the Kronecker symbol δ_{ij} is 1 if $i = j$ and 0 elsewhere. If the size of the texts is counted by the variable z and the occurrences of the words $v_1, \ldots, v_r$ are counted respectively by $x_1, \ldots, x_r$, we get the matrix equation for the generating function of occurrences

$$F(z, x_1, \ldots, x_r) = \mathcal{N}(z) + (x_1\mathcal{R}_1(z), \ldots, x_r\mathcal{R}_r(z))\big(\mathbb{I} - \mathbb{M}(z, x_1, \ldots, x_r)\big)^{-1} \begin{pmatrix} \mathcal{U}_1(z) \\ \vdots \\ \mathcal{U}_r(z) \end{pmatrix}.$$

In this last equation, we have $\mathbb{M}_{ij}(z, x_1, \ldots, x_r) = x_j\mathcal{M}_{ij}(z)$ and the generating functions $\mathcal{R}_i(z)$, $\mathcal{M}_{ij}(z), \mathcal{U}_j(z)$ and $\mathcal{N}(z)$ can be computed explicitly from the set of Equations (2-3).

In particular, when considering the Bernoulli weighted case where the probability of the letters sum up to 1 and a single word w with $\pi_w = \mathbf{P}(w)$, we have the set of equations

$$\mathcal{R}(z) = \frac{\pi_w z^{|w|}}{D(z)}, \quad \mathcal{M}(z) = 1 + \frac{z-1}{D(z)}, \quad \mathcal{U}(z) = \frac{1}{D(z)}, \quad \mathcal{N}(z) = \frac{\mathcal{C}(z)}{D(z)}, \tag{4}$$

where $D(z) = \pi_w z^{|w|} + (1-z)\mathcal{C}(z)$. Finally we get

$$\mathcal{A}^\star = \mathcal{N} + \mathcal{R}\mathcal{M}^\star\mathcal{U} \quad \Longrightarrow \quad F(z,x) = \frac{1}{1 - z + \pi_w z^{|w|}\dfrac{1-x}{x + (1-x)\mathcal{C}(z)}} = \sum_{n,k} f_{n,k} x^k z^n.$$

In this last equation, $f_{n,k}$ is the probability that a text of size n has exactly k occurrences of w.

3.2 Clump analysis for one word

The decomposition of Régnier and Szpankowski is based on a parsing by the occurrences of the considered words. We use a similar approach, but parse with respect to the occurrences of clumps. When they consider the minimal language separating two occurrences, these two occurrences may overlap; in contrast, our approach forbids the overlap of clumps.

A clump of the word w is basically defined as $w\,\mathcal{C}_\circ^\star$, since any element of $\mathcal{C}_\circ$ concatenated to a cluster extends this cluster. In general $\mathcal{C}_\circ^\star$ is ambiguous as can be seen by considering the word $w = aaa$, where we have $\mathcal{C}_\circ = \{a, aa\}$. We can however generate unambiguously $\mathcal{C}_\circ^\star$ as described in the next section.

3.2.1 A prefix code $\mathcal{K}$ to generate unambiguously $\mathcal{C}_\circ^\star$

We refer to Berstel and Perrin [4] for an introduction to prefix codes. See also Berstel [3] for an analysis of counts of words of a pattern $\mathcal{V}$ by semaphore codes $\mathcal{V} - \mathcal{A}^\star\mathcal{V}\mathcal{A}^+$.

We will use the following lemma to derive generating functions.

Lemma 1 *The prefix code $\mathcal{K} = \mathcal{C}_\circ \setminus \mathcal{C}_\circ \mathcal{A}^+$ generates unambiguously the language $\mathcal{C}_\circ^\star$ (i.e., we have $\mathcal{K}^\star = \mathcal{C}_\circ^\star$ as languages).*

Proof: See Figure 1 as illustration of this proof. It is clear that $\mathcal{K}$ is prefix. Let w be the word of which $\mathcal{C}_\circ$ is the strict autocorrelation set. Consider $v \in \mathcal{C}_\circ$. If v does not belong to $\mathcal{K}$, then $v = \kappa_1 v'$ with $\kappa_1 \in \mathcal{K} \subseteq \mathcal{C}_\circ$ and $v' \in \mathcal{A}^+$. Moreover since both v and its prefix κ_1 belong to $\mathcal{C}_\circ$, there exist non-empty words p, p' and s' such that $p = p's'$, $wv = pw$ and $w\kappa_1 = p'w$. Therefore $wv' = s'w$ and $v' \in \mathcal{C}_\circ$. As $|v'| < |v|$; we may iterate the process on the word v'. Since $|v|$ is finite, after a finite number of steps, we get to a decomposition $w = \kappa_1 \dots \kappa_j$ where each κ_i is in $\mathcal{K}$. Since $\mathcal{K}$ is a code, the decomposition of each word of $\mathcal{C}_\circ$ over $\mathcal{K}$ is unique and so is the decomposition of any word of $\mathcal{C}_\circ^\star$. □

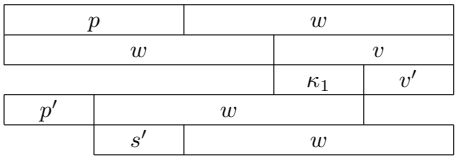

Fig. 1: Proof of Lemma 1.

Moreover, for $c_1, c_2 \in \mathcal{C}_\circ$ and $|c_1| < |c_2|$, the word c_1 is a proper suffix of c_2. From this property we can deduce that there exists non-empty (and non necessarily distinct) words $q_1, q_2, \dots, q_k$ such that $\mathcal{K} = \{\kappa_1, \dots, \kappa_k\}$ can be written $\mathcal{K} = \{q_1, q_2 q_1, \dots, q_k q_{k-1} \dots q_1\}$.

Example 1 *Let $w = abaabaaba$. We have*

$$\begin{array}{r} abaabaaba|\varepsilon \\ abaaba|aba \\ aba|abaaba \\ a|baabaaba \end{array} \quad \Longrightarrow \quad \mathcal{C} = \{\varepsilon, aba, abaaba, baabaaba\} \quad \Longrightarrow \quad \mathcal{K} = \{aba, baabaaba\}.$$

Constructing the prefix-code $\mathcal{K}$. We use the following algorithm, that takes the set of periods (i.e., the lengths of non-empty words in $\mathcal{C}$) as input:

1. start with the word w;
2. shift w to the right to the first self-overlapping position; let κ_1 be the trailing suffix so obtained; insert it in a trie;
3. repeat shifting, obtaining new trailing suffixes; for each new suffix generated, try an insertion in the trie; if you reach a leaf, drop the suffix; elsewhere insert it.

The worst case complexity for this construction is $O(|w|)$, but the average complexity is $O\left(|\mathcal{K}| \times \log |\mathcal{K}|\right)$, the average path length of a trie built over $|\mathcal{K}|$ keys (under the simplifying hypothesis that the words in $\mathcal{K}$ are independent and randomly generated; see Sedgewick and Flajolet [23]).

3.2.2 The language decomposition

Considering the word $w = aaaaa$, we have $\mathcal{C}_\circ = \{a, aa, aaa, aaaa\}$ and $\mathcal{K} = \{a\}$. Moreover, we have $\mathcal{M} = \{a, b(b + ab + aab + aaab + aaaab)^\star aaaaa\}$. We get here $\mathcal{K} \subset \mathcal{M}$ and $\mathcal{M} - \mathcal{K}$ is a set of words ending with w. The languages $\mathcal{M}$ and $\mathcal{K}$ are connected by a simple property that we describe now.

Lemma 2 *For any word w with strict autocorrelation set $\mathcal{C}_\circ$, prefix code $\mathcal{K}$ generating $\mathcal{C}_\circ^\star$ and minimal language $\mathcal{M}$, there exists a non-empty language $\mathcal{L}$ such that*

$$\mathcal{K} \subset \mathcal{M} \qquad \textit{and} \qquad \mathcal{M} - \mathcal{K} = \mathcal{L}w.$$

Proof: We have $\mathcal{K} \subset \mathcal{C}_\circ \subset \mathcal{M}$. If $h \in \mathcal{M} - \mathcal{K}$, we can write $wh = xw$ for some x.

Supposing that $|h| < |w|$, we have $h \in \mathcal{C}_\circ$. Since each word of $\mathcal{C}_\circ$ is decomposable into factors over $\mathcal{K}$ (Lemma 1), if h is not an element of $\mathcal{K}$, there is a previous match with w in $w.h$, and h is not in $\mathcal{M}$; elsewhere h is in $\mathcal{K}$, which is impossible since we considered the set $\mathcal{M} - \mathcal{K}$.

If $|h| = |w|$, we have $h = w \in \mathcal{M}$ and $h \notin \mathcal{K}$ and therefore $w \in \mathcal{M} - \mathcal{K}$. The last possible case is $|x| > |w|$ which implies that $h = y.w$ with $|y| > 0$. □

This leads immediately to the fundamental lemma.

Lemma 3 *The basic equation for the unambiguous combinatorial decomposition of texts on the alphabet $\mathcal{A}$ is*

$$\mathcal{A}^\star = \mathcal{N} + \mathcal{R}w^-(w\mathcal{C}^\star)\big((\mathcal{M} - \mathcal{K})w^-(w\mathcal{C}^\star)\big)^\star\mathcal{U}. \tag{5}$$

Proof: The Equation (5) follows from the parsing of any given text.

Either there is no occurrence of w which means that the text belongs to the "Not" language $\mathcal{N}$.

If there is at least one clump occurrence, we parse as follows: we read until the first occurrence of the clump (a word of $\mathcal{R}$). This occurrence may be followed by any number of overlapping occurrences of w (corresponding to a word of $\mathcal{C}^\star$ and forming a clump). Then we have to wait again the (possible) next occurrence of w, thus reading a word of $\mathcal{M} - \mathcal{K}$. Then a new clump is parsed (corresponding to a word of $\mathcal{C}^\star$). We repeat the last two steps if there are other occurrences of w. Finally we end by reading a word of $\mathcal{U}$ which add no occurrence. □

We can now use the preceding lemma to count several parameters related to the clumps.

3.2.3 Generating functions for the clumps of one word

Let $\Gamma(z, x, \tau)$ be the generating function where the variable x counts the number of occurrences of w in a clump, and the variable τ counts the total number of letters inside clumps; the variable z is used here to count the total length of the texts. We also use a variable γ to count the number of clumps. We have the following theorem.

Theorem 1 *In the weighted model such that $\mathcal{A}(z) = z$ (i.e. where weights of letters of the alphabet $\mathcal{A}$ are probabilities in the Bernoulli model), the generating function counting the number of occurrences of a word w and the number of positions covered by the clumps of w verifies*

$$F(z, \Gamma(z, x, \tau)) = \mathcal{N}(z) + \frac{\mathcal{R}(z)}{\pi_w z^{|w|}}\Gamma(z, x, \tau)\frac{1}{1 - \dfrac{\mathcal{M}(z) - \mathcal{K}(z)}{\pi_w z^{|w|}}\,\Gamma(z, x, \tau)}\mathcal{U}(z), \tag{6}$$

where the generating function of the clumps verifies

$$\Gamma(z,x,\tau) = x\pi_w(z\tau)^{|w|}\frac{1}{1-x\mathcal{K}(z\tau)}. \tag{7}$$

As a consequence, the generating function counting also the number of clumps is

$$G(z,x,\tau,\gamma) = F(z,\gamma\Gamma(z,x,\tau)).$$

Proof: This theorem follows from Lemma 1 and from a direct translation of Equation (5) into generating functions (using the fact that clumps of w correspond to terms $w\mathcal{C}^\star$ in Equation (5)). □

3.2.4 Occurrences of clumps.

By considering $F(z,\gamma\Gamma(z,1,1))$ in Equation (6) and using Equation (7) we obtain the bivariate generating function counting the number of clumps

$$O^{(\mathfrak{K})}(z,\gamma) = \sum_{n,i}\mathbf{P}(O_n^{\mathfrak{K}}=i)\gamma^i z^n = \mathcal{N}(z) + \frac{\gamma\mathcal{R}(z)\mathcal{U}(z)}{1-\gamma\mathcal{M}(z)+(\gamma-1)\mathcal{K}(z)}. \tag{8}$$

We get by differentiation and by using the set of Equations (4)

$$\sum_n \mathbf{E}(O_n^{\mathfrak{K}})z^n = \left.\frac{\partial O^{(\mathfrak{K})}(z,\gamma)}{\partial\gamma}\right|_{\gamma=1} = \frac{\mathcal{R}(z)\mathcal{U}(z)(1-\mathcal{K}(z))}{(1-\mathcal{M}(z))^2} = \frac{\pi_w z^{|w|}(1-\mathcal{K}(z))}{(1-z)^2}.$$

We get similarly by differentiating twice the generating function of the second moment of $O_n^{\mathfrak{K}}$.

We obtain mechanically by Taylor expansions in a neighborhood of $z=1$ the following result.

Proposition 1 (Expectation and variance of the number of clumps – case of one word) *In the Bernoulli model, the expectation and variance of the number of clumps in texts of size n is given by*

$$\begin{aligned}\mathbf{E}(O_n^{\mathfrak{K}}) &= (n-|w|+1)\pi_w(1-\mathcal{K}(1)) - \pi_w\mathcal{K}'(1)\\ \mathbf{Var}(O_n^{\mathfrak{K}}) &= n\times(1-\mathcal{K}(1))^2\mathbf{V}_w - n\times\pi_w(1-\mathcal{K}(1))(\mathcal{K}(1)-2\pi_w\mathcal{K}'(1)) + O(1),\end{aligned}$$

where $\mathbf{V}_w = \pi_w\big(2\mathcal{C}(1)-1-(2|w|-1)\pi_w\big)$ and $\mathcal{K}(z)$ is the generating function of the prefix code generating $\mathcal{C}_\circ^\star$ (see Lemma 1).

This is to compare with the counting of occurrences of a single word w

$$\mathbf{E}(O_n^w) = (n-|w|+1)\pi_w, \qquad \mathbf{Var}(O_n^w) = n\times\mathbf{V}_w + O(1).$$

We obtain as expected smaller expectation and variance for the number of clumps than for the number of word occurrences, with a characteristic reduction coefficient $(1-\mathcal{K}(1))$. We remark here that when $\mathcal{C}=\{\varepsilon\}$ (no autocorrelation), we have $\mathcal{K}(z)=0$ and therefore $\mathbf{E}(O_n^{\mathfrak{K}}) = \mathbf{E}(O_n^w)$ and $\mathbf{Var}(O_n^{\mathfrak{K}}) = \mathbf{Var}(O_n^w)$ as expected.

3.2.5 Occurrences of k-clumps.

By decomposing the equation of a clump of occurrences of w, we can use a formal variable v and write

$$w\mathcal{C}^\star = w + w\mathcal{K} + w\mathcal{K}^2 + \cdots + w\mathcal{K}^{k-2} + vw\mathcal{K}^{k-1} + w\mathcal{K}^k + \dots$$

to count clumps with exactly k occurrences of w.

Denoting $\Gamma_k(z, v)$ the generating function which counts with the variable z the number of letters inside the clumps and where the variable v selects k-clumps, we have

$$\Gamma_k(z, v) = \pi_w z^{|w|} \left(\frac{1}{1 - \mathcal{K}(z)} + (v - 1)\mathcal{K}(z)^{k-1} \right).$$

Substituting this in Equation (6) gives

$$O^{(\mathfrak{K}_k)}(z, v) = \sum \mathbf{P}(O_n^{\mathfrak{K}_k} = i) v^i z^n = F(z, \Gamma_k(z, v)),$$

where $F(z, \Gamma)$ is given by Equation (6).

3.3 Clumps of a finite set of words

We provide in this section a matricial solution for counting clumps of a reduced finite set of words. For sake of simplicity we consider a set of two words $\mathcal{V} = \{v_1, v_2\}$ but our approach is amenable to any reduced finite set.

Similarly to the one word case, we are lead to consider prefix codes generating the correlation of two words. Considering $\mathcal{C}_{ij}^\star$ is not relevant when we have $i \neq j$. However, we can write as previously $\mathcal{K}_{ij} = \mathcal{C}_{ij} - \mathcal{C}_{ij}\mathcal{A}^+$, which defines minimal correlation languages with good properties.

Following a path similar to the proof of Lemma 2, there exists a language $\mathcal{L}$ such that

$$\mathcal{M}_{ij} - \mathcal{K}_{ij} = \mathcal{L} \cdot v_j.$$

We can therefore define the minimal correlation matrix $\mathbb{K}$, the matrix $\mathbb{S} = \mathbb{K}^\star$, and write a clump matrix $\mathbb{G}$ as follows

$$\mathbb{K} = \begin{pmatrix} \mathcal{K}_{11} & \mathcal{K}_{12} \\ \mathcal{K}_{21} & \mathcal{K}_{22} \end{pmatrix}, \qquad \mathbb{S} = \mathbb{K}^\star, \qquad \mathbb{G} = \begin{pmatrix} v_1\mathbb{S}_{11} & v_1\mathbb{S}_{12} \\ v_2\mathbb{S}_{21} & v_2\mathbb{S}_{22} \end{pmatrix}.$$

In this equation, $\mathbb{G}_{ij}$ represents the set of clumps starting with the word v_i and finishing with the word v_j. We obtain now a fundamental matricial decomposition that can be used for further analysis,

$$\mathcal{A}^\star = \mathcal{N} + (\mathcal{R}_1 v_1^-, \mathcal{R}_2 v_2^-)\, \mathbb{G}\Big((\mathbb{M} - \mathbb{K})^- \mathbb{G} \Big)^\star \begin{pmatrix} \mathcal{U}_1 \\ \mathcal{U}_2 \end{pmatrix},$$

where we have $(\mathbb{M} - \mathbb{K})_{ij}^- = (\mathcal{M}_{ij} - \mathcal{K}_{ij}) v_j^-$.

4 Automaton approach

We provide in this section an algorithmic approach by automata for evaluating parameters of clumps for the case of general arbitrary finite sets of words, a question that has never been previously considered in the literature.

For a set $\mathcal{V} = \{v_1, \ldots, v_r\}$ where the right extension set from v_i to v_j is denoted by $\mathcal{E}_{i,j}$ we construct a kind of "Aho-Corasick" automaton on the following set of words X

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

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

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

According to what we want to count or recognize as a language we are led to consider several cases for the set of terminal states:

- *Occurrences of* $v_i \in \mathcal{U}$. We define the set of terminal states T_i as

$$T_i = \mathrm{Pref}(X) \cap \mathcal{A}^\star v_i.$$

- *Occurrences of* $\mathcal{U}$. To count all occurrences, we simply consider the set of terminal states

$$T = \cup_{i=1}^{r} T_i = \mathrm{Pref}(X) \cap \mathcal{A}^\star \mathcal{V}.$$

- *Clumps.* We define the set of final states T_{clumps} in order to accept the language of words ending by the first occurrence of a word in a clump by

$$T_{\text{clumps}} = \mathcal{V} \setminus \mathcal{V}\mathcal{A}^+.$$

Of course this construction does not give in general a minimal automaton. Remark that using different marks for the different sets of terminal states permits to consider simultaneously the different parameters. The automaton is complete and deterministic so that the translation to generating function is straightforward. We can easily derive from this automaton the generating function $F(z, \gamma, \tau, x_1, \ldots, x_r)$ where x_i marks an occurrence of v_i, γ marks the number of clumps, and τ the total number of letters inside the clumps. Indeed, one has to mark some transitions in the adjacency matrix $\mathbb{J}$ according to some simple rules.

- Any transition, labeled by a letter $x \in \mathcal{A}$, is marked by $z\pi(x)$ where $\pi(x)$ is a formal weight for x (usually either the probability of x or 1 if we are considering enumerative generating functions).
- To count occurrences of the v_i's, we mark with the formal variable x_i all transitions going to states in T_i.
- For counting the number of clumps, we mark by γ the transitions going to states in $T_{\text{clumps}} = \mathcal{V} \setminus \mathcal{V}\mathcal{A}^+$, that is states corresponding to first occurrences inside a clump.
- Finally, for the total length covered by clumps, we put a formal weight on the transitions going to a state $p \in T$ by taking into account the number of symbols between the last occurrence of a word of $\mathcal{V}$ and the new one at the end of p. Let us define for a state p (corresponding to a word with an occurrence of some word of $\mathcal{V}$ at the end) the function $\ell(p)$ as the maximal proper prefix q of p in $\mathrm{Pref}(X) \cap \mathcal{A}^\star\mathcal{V}$ if it exists or ε if there is no such prefix. Then we mark all transitions going to such a state $p \in T$ with $\tau^{|p|-|\ell(p)|}$.

We note $z\mathbb{J}(\gamma, \tau, x_1, \ldots, x_r) = z(\mathbb{J}_{i,j})_{1\leq i,j\leq N}$ the transition matrix of the automaton with the previously defined formal labels on transitions, where N is the total number of states of the automaton. Assuming that the initial state has index 1, we get to the generating function that counts the number of clumps and the size of covered positions

$$F(z, \gamma, \tau, x_1, \ldots, x_r) = (1, 0, \ldots, 0)\Big(\mathbf{I} - z\mathbb{J}(\gamma, \tau, x_1, \ldots, x_r)\Big)^{-1}\begin{pmatrix}1\\ \vdots \\ 1\end{pmatrix}. \tag{9}$$

A formal proof of this result (omitted here) relies on the following properties. For a path $s \xrightarrow{h} p$ in the automaton starting at the initial state s and ending on state p after reading a word h, we have: (a) $p \in T_i$ if and only if h ends with an occurrence of v_i; (b) $p \in T_{\text{clumps}}$ if and only if h ends with the first occurrence of $\mathcal{V}$ inside a clump. Additionally, to properly consider the length of a clump, we have to prove that inside a clump, the increase in length of the clump between two occurrences of $\mathcal{V}$ only depends on the state we are reaching.

Note that the multivariate generating function can be obtained by a generalization of the Chomsky-Scützenberger algorithm [5] and that it is possible to transform the automaton constructed for a Bernoulli source to an automaton handling a Markov source of any order (see Nicodème *et al.* [15]).

Examples. Two automata are depicted in Figure 2. The first one (top) corresponds to the set $\mathcal{V} = \{bababa\}$, which gives $\mathcal{E}_u = \{ba, baba\}$ and $X = \{bababa, babababa, bababababa\}$. The second one (bottom) corresponds to the set $\mathcal{V} = \{u_1 = aabaa, u_2 = baab\}$, for which the matrix of right extension sets $\mathcal{E}$ and set X are respectively

$$\mathcal{E} = \begin{pmatrix} baa + abaa & b \\ aa & aab\end{pmatrix}, \text{ and } X = \{aabaa, aabaab, aabaabaa, aabaaabaa, baab, baabaa, baabaab\}.$$

5 Limit laws

We consider here two cases of practical importance.

(a) For clumps of a finite set of words let $\mathbb{J}(\gamma, \tau)$ be the matrix associated to the automaton $\mathcal{T}$ described in the preceding section where γ counts the number of clumps and τ the total size of texts covered by clumps. Since each state $s \in T_{\text{clumps}}$ that recognizes the beginning of a clump is recurrent (that is, here, the final states are always attainable whatever the state we start from), the number of times each of these states is reached in paths of length n (which is equivalent to take the nth power of $\mathbb{J}$) is $\Theta(n)$; we have a finite number of such states and therefore the number of occurrences of clumps is $\Theta(n)$. The corresponding asymptotic normal limit law is a good approximation in the central regime when the size of texts is large.

(b) When the size of the texts is relatively small in comparisons with the size of the words of the pattern, we expect a Poisson law. We give in this section a precise and computable asymptotic Poisson-like approximation for this case when considering clumps of a single word.

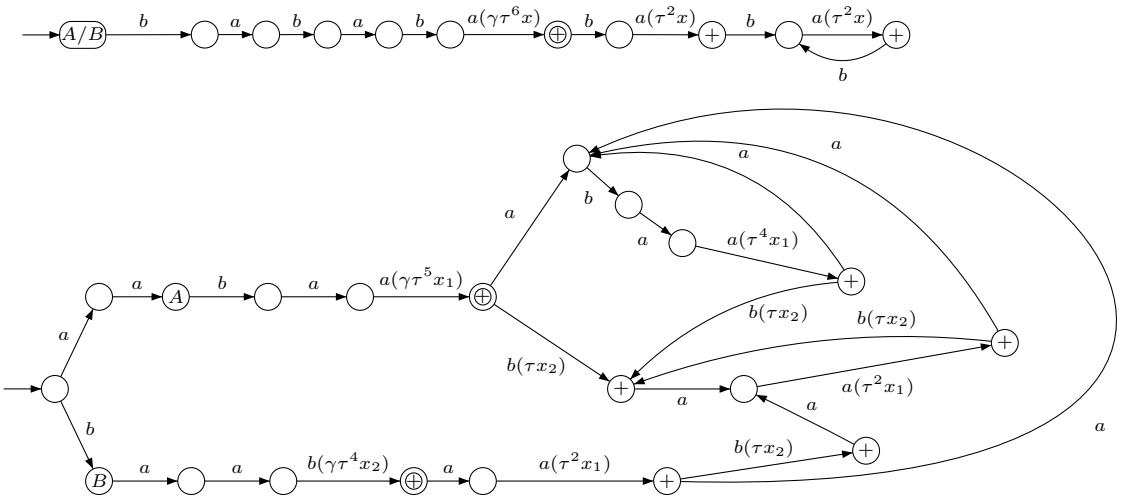

Fig. 2: Two examples of automata, for (top) $\mathcal{V} = \{bababa\}$ and (bottom) $\mathcal{V} = \{v_1 = aabaa, v_2 = baab\}$. The automata are complete and deterministic; however, for sake of clarity, all transitions labeled by a and b ending respectively on state A and B are omitted (which corresponds to the initial state on the first figure). The sign '$+$' indicates that the corresponding prefix (or, equivalently, state) ends with some occurrence of $\mathcal{V}$. The double oval attribute indicates the states where we know that we have entered a new clump. The formal weights on transitions (γ for the number of clumps, τ for the total length of clumps, and x_i's for occurrences of v_i's) are displayed between parenthesis.

The matrix $\mathbb{J}(\gamma, \tau)$ is definite and positive, which entails the existence of a unique positive dominant eigenvalue $\lambda(\gamma, \tau)$ for the matrix (see Gantmacher [7]). We write here $\lambda_\gamma(r) = \lambda(r, 1)$ and $\lambda_\tau(s) = \lambda(1, s)$ (according to the variable we fix to one).

As a consequence, we have the following theorem.

Theorem 2 *Let $O_n^{\mathfrak{K}}$ and $T_n^{\mathfrak{K}}$ be the variables counting respectively the number of clumps and the total number of positions covered by the clumps in random texts of size n. We have as $n \to \infty$.*

– *For clumps of any finite set of words, if $O_n^{\mathfrak{K}} = \Theta(n)$, in the Bernoulli and Markov models, we have*

$$\mathbf{P}\left(\frac{O_n^{\mathfrak{K}} - \mu n}{\sigma\sqrt{n}} \leq y\right) \to \frac{1}{\sqrt{2\pi}} \int_{-\infty}^{y} e^{-t^2/2} dt,$$

where

$$\mu = \mathbf{E}(O_n^{\mathfrak{K}}) = n\lambda'_\gamma(1) + O(1) \quad \textit{and} \quad \sigma^2 = \mathbf{Var}(O_n^{\mathfrak{K}}) = n(\lambda''_\gamma(1) + \lambda'_\gamma(1) - \lambda'_\gamma(1)^2) + O(1).$$

A similar law occurs for the number of covered positions $T_n^{\mathfrak{K}}$ by replacing λ_γ by λ_τ.

– *For clumps of one word (which are not powers of a letter), if $O_n^{\mathfrak{K}} = O(1)$, in the Bernoulli model, there exists $\rho > 1$ and two polynomials $P(z)$ and $Q(z)$, where ρ, $P(z)$ and $Q(z)$ are computable,*

such that,

$$\mathbf{P}(O_n^{\mathfrak{K}} = k) = \frac{\pi_w \rho^{|w|}}{Q(\rho)} \times \frac{1}{k!} \left(\frac{\rho P(\rho) \times n}{(1-\mathcal{K}(\rho))Q(\rho)} \right)^k \times \rho^{-n} \left(1 + O\left(\frac{1}{n}\right) \right). \tag{10}$$

Proof: (sketch)
Normal limit law. We consider the random variable $O_n^{\mathfrak{K}}$. We have from Equation (9)

$$O^{(\mathfrak{K})}(z,r) = F(z,r,1,\ldots,1) = \sum_{n,i} \mathbf{P}(O_n^{\mathfrak{K}} = i) r^i z^n.$$

Since the matrix $\mathbb{J}$ is definite positive, there is a positive real dominant eigenvalue $\lambda_\gamma(r)$. This eigenvalue corresponds to a dominant real positive singularity of order one $1/\lambda_\gamma(r)$ of modulus strictly smaller than the moduli of the other singularities. Applying a Cauchy integral along a circle of radius $R > 1/\lambda_\gamma(r)$ and with an R smaller than the moduli of the other singularities provides an expression

$$\phi_n(r) = [z^n] O^{(\mathfrak{K})}(z,r) = c(r) \big(\lambda_\gamma(r)\big)^n.$$

Using next as n tends to infinity the large powers Theorem of Hwang [11, 12] on $\phi_n(r)$ provides the asymptotic normal law. See Nicodème *et al.* [15] for details.
Poisson law for rare words. We consider a long enough word w (typically of size $\Theta(\log n)$) so that the number of occurrences is $O(1)$. Let $\overline{p}$ be the maximal probability of letters of the alphabet. Taking a Taylor expansion of $O^{(\mathfrak{K})}(z,\gamma)$ in Equation (8) at $\gamma = 0$, and considering the kth Taylor coefficient provide a rational generating function $H_k(z) = [\gamma^k] O^{(\mathfrak{K})}(z,\gamma)$ given by the equation

$$H_k(z) = \frac{\mathcal{R}(z)\mathcal{U}(z)(\mathcal{M}(z) - \mathcal{K}(z))^{k-1}}{(1-\mathcal{K}(z))^k} = \frac{\pi_w z^{|w|} \big(z - 1 + (1-\mathcal{K}(z))D(z)\big)^{k-1}}{(1-\mathcal{K}(z))^k (D(z))^{k+1}}. \tag{11}$$

Following Fayolle [6] and using Rouché theorem, there is a single root ρ of $D(z) = \pi_w z^{|w|} + (1-z)C(z)$ inside the disk $|z| < 1/\overline{p}$. We claim that $(1 - \mathcal{K}(z))$ has no roots inside this disk. If $w \neq \alpha^i$ for $\alpha \in \mathcal{A}$, we have $D(1/\overline{p}) < 0$ for all values of $\overline{p}$. We also have $D(0) = \pi_w > 0$; therefore ρ is real positive.
Writing $D(z) = Q(z)(1 - z/\rho)$ and $P(z) = z - 1 + (1 - \mathcal{K}(z))D(z)$ provides Equation (10). □

This Poisson-like limit has been observed for occurrences of one word by Régnier and Szpankowski [18] in the Bernoulli and Markov models; we conjecture the same result for the count of clumps of one word for rare words in the Markov model.

6 Conclusion

We provided in this article explicit formulas for the generating function counting simultaneously parameters of clumps of reduced set of words; these parameters are the number of occurrences of the clumps, the total size or number of positions of the texts covered by the clumps and the number of occurrences of words of the considered set. We also provide an algorithmic construction by automata that allows the computation of this generating function in the general case of non-reduced sets of words.

An extension of our analysis could lead to a combinatorial analysis of *tandem repeats* or multiple repeats that occur in genomes; large variations of such repeats are characteristic of some genetic diseases.

As mentioned previously, providing explicit expressions in the Markov case for reduced sets requires only some technicalities. On the contrary, finding explicit expressions for parameters of clumps in the non-reduced case remains unsolved.

How does our approach extends to clumps of regular expressions? In this challenging case the star-height theorem implies that we cannot in general find a finite set of words v_i and a finite set of prefix codes $\mathcal{K}_i$ with $1 \leq i \leq \ell$ such that the language $\bigcup_{1\leq i\leq \ell} v_i(\mathcal{K}_i)^\star$ describes the clumps.

References

[1] BARBOUR, A., HOLST, L., AND JANSON, S. *Poisson Approximation*. Oxford University Press, 1992.

[2] BASSINO, F., CLÉMENT, J., FAYOLLE, J., AND NICODÈME, P. Counting occurrences for a finite set of words: an inclusion-exclusion approach. In *Proceedings of the 2007 Conference on Analysis of Algorithms* (2007), P. Jacquet, Ed., DMTCS, proc. AH, pp. 29–44. Proceedings of a colloquium organized at Juan-les-Pins, France, June 2007.

[3] BERSTEL, J. Growth of repetition-free words - A review. *Theoretical Computer Science*, 340 (2005), 280–290.

[4] BERSTEL, J., AND PERRIN, D. *Theory of Codes*. Pure and Applied Mathematics. Academic Press, 1985.

[5] CHOMSKY, N., AND SCHÜTZENBERGER, M. The algebraic theory of context-free languages. *Computer Programming and Formal Languages,* (1963), 118–161. P. Braffort and D. Hirschberg, eds, North Holland.

[6] FAYOLLE, J. An average-case analysis of basic parameters of the suffix tree. In *Mathematics and Computer Science* (2004), M. Drmota, P. Flajolet, D. Gardy, and B. Gittenberger, Eds., Birkhäuser, pp. 217–227. Proceedings of a colloquium organized by TU Wien, Vienna, Austria, September 2004.

[7] GANTMACHER, F. *The theory of matrices. Vols. 1,2*. Encyclopedia of Mathematics. New York: Chelsea Publishing Co. Translated by K. A. Hirsch, 1959.

[8] GOULDEN, I., AND JACKSON, D. *Combinatorial Enumeration*. John Wiley, New York, 1983.

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

[10] GUIBAS, L., AND ODLYZKO, A. Strings overlaps, pattern matching, and non-transitive games. *J. Combin. Theory A*, 30 (1981), 108–203.

[11] HWANG, H.-K. *Théorèmes limites pour les structures combinatoires et les fonctions arithmétiques*. PhD thesis, École polytechnique, Palaiseau, France, Dec. 1994.

[12] HWANG, H.-K. Large deviations for combinatorial distributions I: Central limit theorems. *Ann. in Appl. Probab. 6* (1996), 297–319.

[13] JACQUET, P., AND SZPANKOWSKI, W. Autocorrelation on words and its applications. Analysis of Suffix Trees by String Ruler Approach. *J. Combin. Theory A*, 66 (1994), 237–269.

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

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

[16] PAPE, U. J. *Statistics for Transcription Factor Binding Sites*. PhD thesis, Freie Universität, Berlin, 2008.

[17] RÉGNIER, M. A unified approach to word occurrences probabilities. *Discrete Applied Mathematics 104*, 1 (2000), 259–280. Special issue on Computational Biology.

[18] RÉGNIER, M., AND SZPANKOWSKI, W. On pattern frequency occurrences in a Markovian sequence? *Algorithmica 22*, 4 (1998), 631–649. This paper was presented in part at the 1997 International Symposium on Information Theory, Ulm, Germany.

[19] REINERT, G., AND SCHBATH. Compound Poisson and Poisson process approximations for occurrences of multiple words in Markov chains. *J. Comp. Biol. 5* (1998), 223–253.

[20] REINERT, G., SCHBATH, S., AND WATERMAN, M. Probabilistic and statistical properties of words: an overview. *J. Comp. Biol. 7* (2000), 1–46.

[21] ROQUAIN, E., AND SCHBATH, S. Improved compound poisson approximation for the number of occurrences of multiple words in a stationary markov chain. *Adv. Appl. Prob.*, 39 (2007), 1–13.

[22] SCHBATH, S. Compound Poisson approximation of word counts in DNA sequences. *ESAIM Probab. Statist. 1* (1995), 1–16.

[23] SEDGEWICK, R., AND FLAJOLET, P. *An Introduction to the Analysis of Algorithms*. Addison-Wesley Publishing Company, 1996.

[24] STEFANOV, V., ROBIN, S., AND SCHBATH, S. Waiting times for clumps of patterns and for structured motifs in random sequences. *Discrete Applied Mathematics 155* (2007), 868–880.