Exact p-value calculation for heterotypic clusters of regulatory motifs and its application in computational annotation of cis-regulatory modules

Valentina Boeva, Julien Clément, Mireille Régnier, Mikhail Roytberg, Vsevolod Makeev

Algorithms for Molecular Biology BioMed Central

Research

Open Access

Exact p-value calculation for heterotypic clusters of regulatory motifs and its application in computational annotation of *cis*-regulatory modules

Valentina Boeva*[1,2], Julien Clément[3], Mireille Régnier[2], Mikhail A Roytberg[4,5] and Vsevolod J Makeev[1,6]

Address: [1]Institute of Genetics and Selection of Industrial Microorganisms, GosNIIGenetika, 117545 Moscow, Russia, [2]MIGEC, INRIA Rocquencourt, 78153 Le Chesnay, France, [3]GREYC, CNRS UMR 6072, Laboratoire d'informatique, 14032 Caen, France, [4]Institute of Mathematical Problems of Biology, Russian Academy of Sciences, Puschino, Moscow Region, Russia, [5]Puschino State University, Puschino, Moscow Region, Russia and [6]Engelhardt Institute of Molecular Biology, Russian Academy of Sciences, Moscow, Russia

Email: Valentina Boeva* - valeyo@yandex.ru; Julien Clément - Julien.Clement@info.unicaen.fr; Mireille Régnier - Mireille.Regnier@inria.fr; Mikhail A Roytberg - mroytberg@impb.psn.ru; Vsevolod J Makeev - makeev@genetika.ru

* Corresponding author

Published: 10 October 2007

Algorithms for Molecular Biology 2007, **2**:13 doi:10.1186/1748-7188-2-13

Received: 13 July 2007
Accepted: 10 October 2007

This article is available from: http://www.almob.org/content/2/1/13

Abstract

Background: *cis*-Regulatory modules (CRMs) of eukaryotic genes often contain multiple binding sites for transcription factors. The phenomenon that binding sites form clusters in CRMs is exploited in many algorithms to locate CRMs in a genome. This gives rise to the problem of calculating the statistical significance of the event that multiple sites, recognized by different factors, would be found simultaneously in a text of a fixed length. The main difficulty comes from overlapping occurrences of motifs. So far, no tools have been developed allowing the computation of *p*-values for simultaneous occurrences of different motifs which can overlap.

Results: We developed and implemented an algorithm computing the *p*-value that *s* different motifs occur respectively k_1, ..., k_s or more times, possibly overlapping, in a random text. Motifs can be represented with a majority of popular motif models, but in all cases, without indels. Zero or first order Markov chains can be adopted as a model for the random text. The computational tool was tested on the set of *cis*-regulatory modules involved in *D. melanogaster* early development, for which there exists an annotation of binding sites for transcription factors. Our test allowed us to correctly identify transcription factors cooperatively/competitively binding to DNA.

Method: The algorithm that precisely computes the probability of simultaneous motif occurrences is inspired by the Aho-Corasick automaton and employs a prefix tree together with a transition function. The algorithm runs with the $O(n|\Sigma|(m|\mathcal{H}| + K|\sigma|^K)\prod_i k_i)$ time complexity, where n is the length of the text, $|\Sigma|$ is the alphabet size, m is the maximal motif length, $|\mathcal{H}|$ is the total number of words in motifs, K is the order of Markov model, and k_i is the number of occurrences of the ith motif.

Conclusion: The primary objective of the program is to assess the likelihood that a given DNA segment is CRM regulated with a known set of regulatory factors. In addition, the program can also

be used to select the appropriate threshold for PWM scanning. Another application is assessing similarity of different motifs.

Availability: Project web page, stand-alone version and documentation can be found at http://bioinform.genetika.ru/AhoPro/

Background

During the past few years, a number of computational tools have been designed [1-3] for locating potential *transcription factor binding sites* (TFBSs) in nucleotide sequences, e.g., in compilations of sequences upstream of putative co-regulated genes. In parallel, experimental approaches were developed [4], which allowed identification of binding motifs for many different transcription factors. Experimental [5] and bioinformatical [6] studies demonstrated that sequences of regulatory DNA that bind transcription factors can exhibit many different types of architecture. In eukaryotes TFBSs found in DNA sequences often form rather dense clusters: this was demonstrated both by experimental [5,7] and computational [8,9] methods. Such clusters can contain sites binding the same factor or several different factors [10]. The *cis*-regulatory module (CRM) in this case contains respectively homotypic or heterotypic clusters of motifs specifically recognized by binding proteins [11].

The particular arrangement of motifs in a homotypic or heterotypic cluster is not random, and it is commonly accepted, that the motif arrangement within a CRM is important for its functionality [12-20]. Bioinformatics studies indicate that antagonistic factors often bind to overlapping sites [21] whereas synergetic factors are often positioned within a fixed distance [20], often close to the multiple of 10.2 bp, the DNA double-helix pitch value [21].

Non-random arrangements of TFBSs within regulatory segments of DNA sequences are exploited in several TFBS identification tools, and it was observed that cooperativity-based discrimination of TFBSs surpasses the performance of models for individual TFBSs [22].

On observing a cluster of TFBSs in some genome segment one can calculate the probability of observing similar site arrangements in a random sequence. This idea of evaluating the statistical significance of heterotypic clusters of sites was implemented in many programs including ClusterDraw [23], ModuleSearcher [24], MCAST [25], eCIS-ANALYST [26], Cister [27], Cluster-Buster [28] and TargetExplorer [29]. At the moment, such programs use empirical procedures like motif counting in biological and simulated sequences to assess the significance of observed site clustering. But it is highly desirable to have a good statistical measure of site clustering, and we believe that the best measure is the *p*-value of obtaining the observed cluster by chance in a random sequence of a Markov or Bernoulli (common name for Markov chain of order 0) type. In the case of heterotypic clusters one needs to take into account possible overlapping occurrences of different motifs, a problem that was considered difficult until now [30]. In the case of homotypic clusters, an approximate statistical scoring function was constructed [8,31]; this approach has been implemented in algorithms like FLY-ENHANCER [32], SCORE [33], and CLUSTER [34]. However, this approximation performs poorly for highly overlapping TFBSs. One cannot ignore site overlapping if the motifs are fuzzy (highly degenerate), which is often the case for so-called "shadow sites" [31]. In the case of heterotypic clusters, competing factors can bind even to very well determined motifs that overlap.

Representation of protein binding motifs in nucleotide sequences

Experimental methods on protein binding to DNA usually locate some DNA segment, or word in DNA text, as a probable binding target. Proteins can bind to similar DNA words [4], the whole assembly of which can be called a motif. The simplest motif representation is the enumeration of sequences that can be bound by a transcription factor (TF) [35]. Sometimes, information about binding sites can be found in SELEX [36,37] or Protein Binding Microarray (PBM) experiments [38]. However, it is possible that such experiments do not give the exhaustive list of sequences of binding sites, so one needs to expand the list of putative binding sites using an appropriate criterion, which brings about the problem of the generalization of several known examples.

For instance, several words aligned with mismatches, can be generalized to IUPAC string (like RSTGACTNMNW for AP-1 binding sites [39]) by disregarding correlated substitutions in different motif positions [40]. Another example of generalization is the set of words that can deviate from a consensus word for less than a given number of mismatches.

The most popular way to represent binding sites is a Position Weight Matrix (PWM), which is also called position-specific weight matrix (PSWM) or position-specific scoring matrix (PSSM) [41]. For a text with length D over an alphabet Σ with $|\Sigma|$ symbols, a PWM is a $|\Sigma| \times D$ matrix:

each row corresponding to a symbol of the alphabet Σ, and each column to a position in the motif. For DNA texts, one has Σ = {*A*, *C*, *G*, *T*}. The PWM score is defined as $\sum_{i=1}^{L} m_{\omega(i),i}$, where *i* represents a position in the *D*-substring, $\omega(i)$ the symbol at position *i* in the substring, and $m_{\alpha,i}$ the score in row α, column *i* of the matrix. So, given a cutoff value, one gets a list of *D*-sequences that score higher than this cutoff; thus representing possible DNA binding sites for the protein.

Any of the three motif representations above can be converted to a list of words. The same is true for many other representations of motifs. In this study, we consider only the motifs that can be represented as a set of words.

P-value for clusters of motif occurrences, problem formulation

The objective of this work is to develop a statistical criterion to assess clustering of TFBS. Intuitively, a TFBS cluster is a DNA segment simultaneously containing "too many" TFBSs for given factor proteins; such a segment can often operate as a CRM regulated by these TFs. From a formal point of view, the problem we address here is as follows. Let *s* sets of words $\mathcal{H}_1,...,\mathcal{H}_s$ be given. Typically, each set $\mathcal{H}_i$ is associated to a TF motif. Given a *s*-tuple of integers (k_1, ..., k_s), we compute the corresponding *p*-value, that is the probability to find at least k_i occurrences of words from each set $\mathcal{H}_i$ in a random text of size *n*. We assume that the texts where motifs are searched are randomly generated by a Bernoulli process or a Markov model of order *K*. If (k_1, ..., k_s) occurrences of motifs $\mathcal{H}_1,...,\mathcal{H}_s$ are found in a DNA segment, the *p*-value can be used to infer if such numbers of occurrences could be found by chance.

Related work

Most previous works address counting problems for one set of several words $\mathcal{H}$. In contrast, in this paper we deal with a separate counting for several sets of several words $\mathcal{H}_1,...,\mathcal{H}_s$, each set $\mathcal{H}_j$ represents one TFBS motif.

All methods of solving the problem of *p*-value calculations for multiple occurrences of words from a set $\mathcal{H}$ study some basic languages. Let $L_n(\mathcal{H}; k)$ be the set of texts of length *n* containing at least *k* occurrences of $\mathcal{H}$. The desired *p*-value would therefore be the probability $\mathbf{P}(L_n(\mathcal{H}; k))$. Let $\mathcal{R}^k_{\mathcal{H}}$ be the set of texts of all lengths that contain exactly *k* words of $\mathcal{H}$, the last one occurring as a suffix [42]. For any H_j in $\mathcal{H}$, let $\mathcal{R}^k_{H_j}$ be the subset of $\mathcal{R}^k_{\mathcal{H}}$ where H_j is a suffix. One observes that a text contains at least *k* occurrences if and only if it admits a prefix in $\mathcal{R}^k_{\mathcal{H}} = \bigcup_{H_j \in \mathcal{H}} \mathcal{R}^k_{H_j}$. One defines $r_j^k(p)$ as the probability that a text of size *p* be in set $\mathcal{R}^k_{H_j}$. If no word in $\mathcal{H}$ is a subword of another word in $\mathcal{H}$, the probability $\mathbf{P}(L_n(\mathcal{H}; k))$ to find at least *k* occurrences of words from $\mathcal{H}$ in a random text of length *n* satisfies

$$\mathbf{P}(L_n(\mathcal{H};k)) = \sum_{p \le n} \sum_{H_j \in \mathcal{H}} r_j^k(p)$$

Therefore, one tries to compute the sequence of $(r_j^k(p))$ values.

Linear induction

In the first class of methods [43-46], one computes, implicitly or explicitly, probabilities $\mathbf{P}(L_n(\mathcal{H}; k))$ up to a given text length *n*. Such methods are intrinsically linear in *n*. In [43-46] one relies on a recurrence relation on $r_j^k(n)$ that extends the one originally given in [47]. Typically, one step will cost $O(|\mathcal{H}|m)$, where $\mathcal{H}$ is a set of words of length *m* and $|\mathcal{H}|$ is its cardinality. Time complexity is $O(n|\mathcal{H}|m)$ and, relying on a combinatorial property, [44] achieves optimal space complexity $O(|\mathcal{H}| \log |\mathcal{H}|m)$. However the authors of [44] do not consider several motifs occurrences and restrict themselves to the Bernoulli model. The authors of [43] consider the Markov model, still using one motif for TFBS.

Algebraic Formulae

In a second class of methods [47-52], a preprocessing computes *generating functions*

$$r_j^k(z) = \sum_n r_j^k(n) z^n.$$

In a second step, probabilities $\mathbf{P}(L_n(\mathcal{H}; k))$ are either extracted from the generating function or approximated.

In [49,53], $r_j^k(z)$ are the solutions of a system of equations. To derive these equations, the authors build an

automaton that recognizes these languages $\mathcal{R}^k_{\mathrm{H}_j}$ (one can prove that they are regular).

A language approach [50] or an induction [48] leads to a formal expression that depends on the words overlaps. The main drawback is that these methods need to compute the determinant of a matrix of polynomials with a huge dimension, e.g. $O(|\mathcal{H}|)$. This $O(|\mathcal{H}|^2)$ *symbolic computation* may be more expensive than the extraction step or the linear computation above, that involve *arithmetic operations* on real numbers.

When the preprocessing step is achievable, the extraction step is amenable to the solution of a linear recurrence of degree $m|\mathcal{H}|$; therefore, its complexity is $O(m|\mathcal{H}|n)$ and a classical optimization yields $O(m|\mathcal{H}|\log n)$. There exists some good implementations that are numerically stable. One may cite the REGEXPCOUNT [54] or EXCEP [55] programs that rely on Fast Fourier Transform.

Finally, approximations are available, the computation of which is constant with respect to n, but not to $\mathcal{H}$. One approach is the compound Poisson approximation [56], but this approximation is not precise enough [57]. Asymptotic results can also be derived from the algebraic formulae above [44,58], not needing an explicit expression for $r^k_j(z)$, and therefore avoiding the expensive determinant computation. Time complexity, typically, is the one for computing all possible overlaps, that is approximately $O(|\mathcal{H}|^2)$. This yields extremely precise results when the expectation of the number of occurrences, $nP(\mathrm{H})$ is very small [59] or close to 1 [51] (the case studied the most often). Case $nP(\mathrm{H}) \sim 2$ is achieved in [60]. Nevertheless, extension to larger values of k or multioccurrences and multisets is still open.

Methods

Here we consider in detail the approach we suggest.

A motif assigned to a TF is a finite set of words $\mathcal{H} = (\mathrm{H}_1, ..., \mathrm{H}_r)$ where each word represents one putative TF binding site in DNA. Note that words in motif can generally be of different lengths. However, no word from $\mathcal{H}$ can contain another word from $\mathcal{H}$ as a substring. We consider, as an occurrence of motif $\mathcal{H}$ in text T, any occurrence of any word $\mathcal{H}_j \in \mathcal{H}$ in T. Below all texts and words in motifs are sequences on a given alphabet Σ.

Let $(\mathcal{H}_1,...,\mathcal{H}_s)$ be s different motifs. Our objective is to calculate the probability (p-value) that motifs $(\mathcal{H}_1,...,\mathcal{H}_s)$ have respectively at least $(k_1, ..., k_s)$ possibly overlapping occurrences in a random text T_n.

To be more precise, there is a probability distribution defined on the set Σ^n of all texts of length n in the alphabet Σ; the most widely used models are random Bernoulli trials and a Markov model of order K. Denote as $L_n(\mathcal{H}_1,...,\mathcal{H}_s; k_1, ..., k_s)$ the set of all texts of length n containing at least k_i possibly overlapping occurrences of each motif $\mathcal{H}_i$; $i = 1, ..., s$. Then the desired p-value is the probability $\mathbf{P}(L_n(\mathcal{H}_1,...,\mathcal{H}_s; k_1, ..., k_s))$ of the set $L_n(\mathcal{H}_1,...,\mathcal{H}_s; k_1, ..., k_s)$ with respect to the given probability distribution on Σ^n.

Our approach to the calculation of this p-value is similar to that published in [61], which was used there to calculate seed sensitivity in local alignment search. The approach exploits the fact that the algorithm of Aho and Corasick [62] can be modified to efficiently determine whether a given text belongs to the set $L_n(\mathcal{H}_1,...,\mathcal{H}_s; k_1, ..., k_s)$ or not. Ideas published in [61] and [62] can be adopted to compute the probability $\mathbf{P}(L_n(\mathcal{H}_1,...,\mathcal{H}_s; k_1, ..., k_s))$ that the random text $T_n \in \Sigma^n$ belongs to the set $L_n(\mathcal{H}_1,...,\mathcal{H}_s; k_1, ..., k_s)$.

We start from the simplest case of one motif $\mathcal{H}$ for which we calculate the probability $\mathbf{P}(L_n(\mathcal{H}; 1))$ that text T_n contains at least one occurrence of the motif with respect to a Bernoulli probability distribution. More complicated cases (arbitrary number of occurrences; arbitrary number of motifs; Markov distribution) will be discussed in the following sections.

Construction of Aho-Corasick traversal

Aho and Corasick [62] have proposed the algorithm determining if a given text T contains an occurrence of a word from a given set $\mathcal{H}$. The basic data structure is a prefix tree which is a variant of the classical trie $\mathcal{T}(\mathcal{H})$ [42] that may be built on the set of words $\mathcal{H}$. Let $Q_{\mathcal{H}}$ denote the set of prefixes of these words. In the following, we identify a word $q \in Q_{\mathcal{H}}$ with node $Node(q)$ at the end of the branch labeled by q. In particular, the root is identified

with the empty string ε. The length of a prefix is the depth of *Node* (q).

The classic Aho-Corasick algorithm is a tree traversal determined by a *transition function* $\delta : Q_{\mathcal{H}} \times \Sigma \to Q_{\mathcal{H}}$ defined as follows. For any pair (p, a) in $Q_{\mathcal{H}} \times \Sigma$, $\delta(p, a)$ is the largest suffix of concatenation pa that belongs to $Q_{\mathcal{H}}$. Remark that $\delta(p, a) = pa$ iff $pa \in Q_{\mathcal{H}}$.

Given a text T read from left to right, let $T[i]$ denote the letter of T at position i. Let q_i be the largest suffix in text $T[1] \cup T[i]$ that belongs to $Q_{\mathcal{H}}$. The sequence of nodes visited during the traversal are defined by words q_i that satisfy the inductive relationship

$$\forall i \geq 0,\ q_{i+1} = \delta(q_i, T[i+1]),$$

with the initial condition $q_0 = \varepsilon$.

Example: Let $\mathcal{H}$ be the set {AAA, AAC, ACA, ACA, CCT}. The corresponding tree $\mathcal{T}(\mathcal{H})$ is depicted in Figure 1. Values of δ function are given in Table 1. Aho-Corasick traversal of tree $\mathcal{T}(\mathcal{H})$ according to text T = 'ATGCCAACCTT' produces the following sequence of nodes $\{q_i\}_{i \geq 1}$ in $Q_{\mathcal{H}}$ (the numbers of corresponding nodes in Figure 1 are shown in square brackets): A[1], ε[0], ε[0], C[2], CC[5], A[1], AA[3], AAC[7], ACC[9], CCT[10], ε[0].

$\mathcal{T}(\mathcal{H})$ and transition function δ can be efficiently constructed with an algorithm proposed by Aho and Corasick

Table 1: Values of δ function for the set $\mathcal{H}$ = {aaa, aac, aca, acc, cct}.

$q\backslash\alpha$	A	C	G	T
0	1	2	0	0
1	3	4	0	0
2	1	5	0	0
3	6	7	0	0
4	8	9	0	0
5	1	5	0	10
6	6	7	0	0
7	8	9	0	0
8	3	4	0	0
9	1	5	0	10
10	1	2	0	0

Values of $\delta(q, \alpha)$ function for $q \in Q$ and α = *A*, *C*, *G*, *T* constructed for the set $\mathcal{H}$ = {AAA, AAC, ACA, ACC, CCT}.

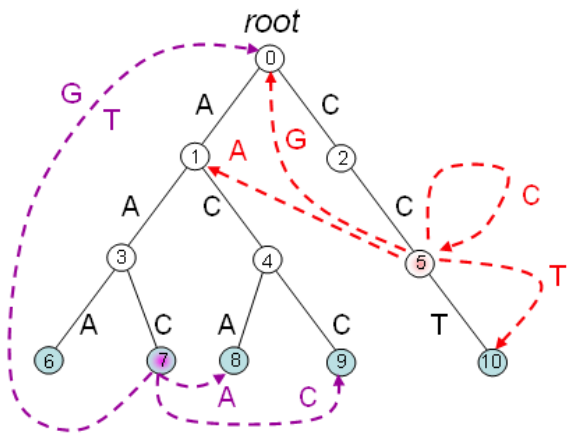

Figure 1

Tree $\mathcal{T}(\mathcal{H})$ for the set $\mathcal{H}$ = {aaa, aac, aca, acc, cct} with dashed links for δ function. Tree $\mathcal{T}(\mathcal{H})$ for the set $\mathcal{H}$ = {AAA, AAC, ACA, ACC, CCT}. Dashed colored links represent δ function for internal node (5) – in red, and for marked node (7) corresponding to the word AAC $\in \mathcal{H}$ – in purple.

[62]. Both time and space of the algorithm is proportional to the sum of lengths of all words from $\mathcal{H}$.

The combination of tree $\mathcal{T}(\mathcal{H})$ and transition function δ allows solving numerous pattern matching problems: search of the first occurrence of a word from a given set, search of all occurrences, word counting, *etc.*

Bernoulli text model. Probability to find at least one occurrence of a single motif

In this section we consider the simplest case. One computes the *p*-value for a single motif in a text T_n of length n, assuming that T_n is generated by independent Bernoulli random trials over alphabet Σ. The algorithm computes probabilities $\mathbf{P}(L_n(\mathcal{H}; 1))$ by induction on n.

To describe the algorithm we divide the set Σ^i of all texts T_i of length i into classes that do and do not contain occurrences of $\mathcal{H}$.

Definition 1 *A text T_i belongs to class $C_i(0; q)$ iff*

1. Length of T_i is i,

2. T_i does not contain words from $\mathcal{H}$,

3. A traversal AC ($\mathcal{T}(\mathcal{H})$, T_i) *ends at node q.*

A text T_i *belongs to class* $G_i(1)$ *iff*

(i) Length of T_i *is i,*

(ii) T_i *does contain at least one occurrence of a word from* $\mathcal{H}$.

For a given number i larger than m, the union for classes $C_i(0; q)$, where q is in $Q_{\mathcal{H}} \setminus \mathcal{H}$ and the class $G_i(1)$ form a partition of the set Σ^i of all texts of length i, i.e., any texts of length i belongs either to a class $C_i(0; q)$ for some q in $Q_{\mathcal{H}} \setminus \mathcal{H}$, or to a class $G_i(1)$. Indeed, condition 3. means that the largest suffix of T_i in $Q_{\mathcal{H}}$ is q. It follows from condition 2. that classes $C_i(q; 0)$ are empty if q is in $\mathcal{H}$. A text T_i of length i is in $G_i(1)$ if and only if a node of $\mathcal{H}$ was visited during the traversal.

Let $\mathbf{P}(C_n(0; q))$ and $\mathbf{P}(G_n(1))$ denote probabilities that a text T_n belongs to class $C_n(0; q)$ and $G_n(1)$, respectively. Then, $L_n(\mathcal{H}; 1) = G_n(1)$; therefore the desired p-value $\mathbf{P}(L_n(\mathcal{H}; 1))$ is equal to $\mathbf{P}(G_n(1))$.

The algorithm calculates probabilities $\mathbf{P}(C_i(0; q))$ and $\mathbf{P}(G_i(1))$ using induction on length i. For $i = 0$, these probabilities obviously comply with: $\mathbf{P}(C_0(0; \varepsilon)) = 1$; $\mathbf{P}(C_0(0; q)) = 0$, for any $q \neq \varepsilon$; $\mathbf{P}(G_0(1)) = 0$.

The values of $\mathbf{P}(C_{i+1}(0; q))$ and $\mathbf{P}(G_{i+1}(1))$ are calculated using values of $\mathbf{P}(C_i(0; q))$ and $\mathbf{P}(G_i(1))$. Therefore, the needed space is proportional to the size of $Q_{\mathcal{H}}$ (see section *Extensions and complexity* below).

Calculation of values $\mathbf{P}(C_{i+1}(0; q))$ and $\mathbf{P}(G_{i+1}(1))$ is based on the following observations. Let U be a set of texts of the same length over the alphabet Σ, $\mathbf{P}(U)$ the probability of U in the Bernoulli model and a a character in Σ. Let $U \cdot a$ be the set of all possible concatenations, i.e., $U \cdot a = \{xa | x \in U\}$. And in the case of the Bernoulli model

$$\mathbf{P}(U \cdot a) = \mathbf{P}(U)\,\mathbf{P}(a). \quad (1)$$

Then the following relations hold for any $i \in \{1, ..., n - 1\}$ and Σ:

(i) if the text T_i contains a word from $\mathcal{H}$ then all its concatenations with characters from Σ would contain a word from $\mathcal{H}$; i.e.,

$$G_i(1) \cdot a \subset G_{i+1}(1). \quad (2)$$

(ii) if the text T_i does not contain a word from $\mathcal{H}$ and belongs to $C_{i+1}(0; q)$, i.e., ends with $q \in Q_{\mathcal{H}} \setminus \mathcal{H}$, then its concatenation $T_i \cdot a$ belongs to the class determined by the result of the Aho-Corasick transition function $\delta(q, a)$; i.e.,

$$\text{if } \delta(q, a) \in \mathcal{H}, \quad \text{then } C_i(0; q) \cdot a \subset C_{i+1}(0; \delta(q, a)) \quad (3)$$

$$\text{otherwise} \quad C_i(0; q) \subset G_{i+1}(1). \quad (4)$$

Remembering that classes $C_i(0; q)$ for different q and $G_i(1)$ form a partition of Σ^i, we obtain the following relation for the texts containing words from $\mathcal{H}$:

$$G_{i+1}(1) = \{\bigcup_{a \in \Sigma} G_i(1) \cdot a\} \cup \{\bigcup_{(q,a);\delta(q,a) \in \mathcal{H}} C_i(0; q) \cdot a\}. \quad (5)$$

Similarly, classes of texts that do not contain words from $\mathcal{H}$ satisfy

$$\forall q' \in Q_{\mathcal{H}} \setminus \mathcal{H}: \quad C_{i+1}(0; q') = \bigcup_{(q,a);\delta(q,a)=q'} C_i(0; q) \cdot a. \quad (6)$$

Classes $C_i(0; q)$ for different q in $Q_{\mathcal{H}} \setminus \mathcal{H}$ and $G_i(1)$ form a partition of Σ^i; classes $C_i(0; q)$ are empty if q is in $\mathcal{H}$. Relations (5) and (6) with the help of (1) yield the recursive expressions for probabilities $\mathbf{P}(C_{i+i}(0; q))$ and $\mathbf{P}(G_{i+1}(1))$ in the Bernoulli case:

$$\mathbf{P}(G_{i+1}(1)) = \mathbf{P}(G_i(1)) + \sum_{(q,a):\delta(q,a) \in \mathcal{H}} \mathbf{P}(C_i(0; q)) \cdot p(a), \quad (7)$$

$$\mathbf{P}(C_{i+1}(0; q')) = \sum_{(q,a):\delta(q,a)=q'} \mathbf{P}(C_i(0; q)) \cdot p(a). \quad (8)$$

The run-time for each step of the computation of $C_{i+1}(0; q)$ and $G_{i+1}(1)$ is $O(|Q_{\mathcal{H}}| \cdot |\Sigma|)$; therefore the total time of all n stages of p-value computation is $O(|Q_{\mathcal{H}}| \cdot |\Sigma| \cdot n)$.

The approach described in this section can be readily extended to the case of multiple occurrences of motif $\mathcal{H}$. The detailed procedure can be found in Additional file 1.

Bernoulli text model. Probability to find multiple occurrences of multiple motifs

DNA transcription is usually regulated with several factors simultaneously interacting with DNA and specifically recognizing different DNA sites. Individual regulatory segment of DNA can contain many binding sites for several factors, often substantially overlapping with each other [5]. This brings about a problem of studying of co-occurring motifs.

Let $(\mathcal{H}_1,...,\mathcal{H}_s)$ be s different motifs. Our objective is to calculate the probability that motifs $(\mathcal{H}_1,...,\mathcal{H}_s)$ have respectively at least $(k_1, ..., k_s)$ possibly overlapping occurrences in the random text T_n of the length n. This p-value is the probability $\mathbf{P}\,(L_n(\mathcal{H}_1,...,\mathcal{H}_s; k_1, ..., k_s))$ to obtain text T_n belonging to the set of texts $L_n(\mathcal{H}_1,...,\mathcal{H}_s; k_1, ..., k_s)$. In this section, we will suppose that the probability of each text is given by Bernoulli model. The Markov case will be considered in the next subsection. The recursion for multiple occurrences of multiple motifs obtained here is rather tricky. Therefore we suggest the reader to see Additional file 1 where we describe the recursion for the simpler case of multiple occurrences of a single motif

Let us consider the union $\mathcal{H}$ of individual motifs $\mathcal{H}=\mathcal{H}_1\cup\cdots\cup\mathcal{H}_s$. It contains all words that belong to any of motifs $\mathcal{H}_i$. The tree $\mathcal{T}(\mathcal{H})$ is constructed for the overall set $\mathcal{H}$, its nodes $Q_{\mathcal{H}}$ contain all possible prefixes of all motifs from $(\mathcal{H}_1,...,\mathcal{H}_s)$. A node of the tree $q \in Q_{\mathcal{H}}$ can belong to some motif $\mathcal{H}_k$ or simultaneously to several different motifs from $\{\mathcal{H}_j\}_{1\le j\le s}$. Let each node $q \in Q_{\mathcal{H}}$ be marked with numbers j of motifs $\mathcal{H}_j$ to which it belongs. Nodes, corresponding to proper prefixes of $\mathcal{H}$, remain unmarked. The transition function $\delta : Q_{\mathcal{H}} \times \Sigma \to Q_{\mathcal{H}}$ is defined as it was defined in the case of a single motif for the unified motif $\mathcal{H}$.

All texts T_n of length n are classified into classes depending on occurrences of different $\mathcal{H}_j$. In this case it is difficult to introduce the target class G, since when the target number of occurrences k_i is attained for some motif $\mathcal{H}_i$, the corresponding value k_j may not yet be attained for another motif $\mathcal{H}_j$. Therefore we need to introduce the occurrence index of a set of motifs.

Definition 2 *Let the target number of occurrences of motif $\mathcal{H}_i$ be k_i. Then, the occurrence index $\Lambda_{(k_1,...,k_s)}(l_1, ..., l_s)$ of a set of motifs $(\mathcal{H})$ in the text T_n containing l_i possibly overlapping occurrences of each $\mathcal{H}_i$ is an s-vector the ith component of which can be calculated as follows*:

$$[\Lambda_{(k_1,...,k_s)}(l_1,...,l_s)]_i = \lambda_i = \begin{cases} l_i & if \quad l_i \le k_i, \\ k_i & if \quad l_i > k_i. \end{cases} \tag{9}$$

Definition 3 *A text T_i belongs to class $C_i(\lambda_1, ..., \lambda_s; q)$, $0 \le \lambda_i \le k_i$ iff*

1. Length of T_i equals i,

2. The occurrence index of motifs $(\mathcal{H}_1,...,\mathcal{H}_s)$ in text T_i is equal to $(\lambda_1, ..., \lambda_s)$,

3. A traversal AC $(\mathcal{T}(\mathcal{H}), T_i)$ ends in node q.

A text T_i belongs to class $G_i(k_1, ..., k_s)$ if it belongs to the union of classes

$$G_i(k_1,...,k_s) = \bigcup_{q\in\mathcal{H}} C_i(k_1,...,k_s;q). \tag{10}$$

The desired p-value $\mathbf{P}\,(L_n(\mathcal{H}_1,...,\mathcal{H}_s; k_1, ..., k_s))$ is equal to $\mathbf{P}\,(G_n(k_1, ..., k_s))$. The value is calculated iteratively. Again, we have a sum over all possible tree nodes q and symbols a. Now, q', the image of the transition function $\delta(q, a)$ can belong simultaneously to several motifs $\{\mathcal{H}_j\}_{1\le j\le s}$. Thus, the resulting probability $\mathbf{P}\,(C_{i+1}(\lambda_1, ..., \lambda_s; q'))$ that text T_{i+1} belongs to class $C_{i+1}(\lambda_1, ..., \lambda_s; q')$ calculates as

$$\mathbf{P}(C_{i+1}(\lambda_1,...,\lambda_s;q')) = \sum_{(q,a):\delta(q,a)=q'} \sum_{(r_1,...,r_s)\in\mathbf{J}} \mathbf{P}(C_i(r_1,...,r_s;q))\cdot p(a) \tag{11}$$

where the summation in the second sum is performed over all allowed s-tuples of indexes $(r_1, ..., r_s)$ which together make the set of s-tuples **J**. A s-tuple of indexes $(r_1, ..., r_s)$ belongs to **J** if it complies with the following conditions:

1. if $q' \notin \mathcal{H}_j$ then $r_j = \lambda_j$,

2. if $q' \mathcal{H}_j$ and $\lambda_j < k_j$ then $r_j = \lambda_j - 1$,

3. if $q' \in \mathcal{H}_j$ and $\lambda_j = k_j$ then $r_j = k_j$ or $r_j = k_j - 1$.

Implementation details

Our basic data structure is the prefix tree; we use its standard representation [42] [see also Additional files 2 and 3 for *Tree construction from PWM motif representation*]. Each tree node $q \in Q_{\mathcal{H}}$ is supplied with several additional variables.

At stage $(i + 1)$ of probability computation the values $\mathbf{P}(C_{i+1}(\lambda_1, ..., \lambda_s; q))$ become computed from the values $\mathbf{P}(C_i(\lambda_1, ..., \lambda_s; q))$ obtained at the previous stage of induction. Therefore, at stage $(i + 1)$, one no longer needs the values calculated at stage $(i - 1)$. Thus, each node is supplied with two $k_1 \times \cup \times k_s$-arrays of real values $\mathbf{C_0}$ and $\mathbf{C_1}$ for storing $\mathbf{P}(C_i(\lambda_1, ..., \lambda_s; q))$ and $\mathbf{P}(C_{i+1}(\lambda_1, ..., \lambda_s; q))$ for different λ_j. $\mathbf{C_0}$ is used to store probabilities for even text lengths while $\mathbf{C_1}$ for odd.

In implementation the calculation of values $\mathbf{P}(C_{i+1}(\lambda_1, ..., \lambda_s; q'))$ from $\mathbf{P}(C_i(\lambda_1, ..., \lambda_s; q))$ for all $q', q \in Q_{\mathcal{H}}$ and $(\lambda_1, ..., \lambda_s)$: $0 \le \lambda_j \le k_j$, $1 \le j \le s$, is performed in the parallel way. Initially we set all the values $\mathbf{P}(C_{i+1}(\lambda_1, ..., \lambda_s; q'))$ to 0. Then we look over all tuples $(r_1, ..., r_s; q)$, where $q \in Q_{\mathcal{H}}$ and $(r_1, ..., r_s)$: $0 \le r_j \le k_j$, $1 \le j \le s$. For each tuple $(r_1, ..., r_s; q)$ and all letters $a \in \Sigma$ we find the prefix $q' = \delta(q, a)$ and the value $\mathbf{P}(C_i(r_1, ..., r_s; q)) \cdot p(a)$. Then we add $\mathbf{P}(C_i(r_1, ..., r_s; q)) \cdot p(a)$ to the value $\mathbf{P}(C_{i+1}(\lambda_1, ..., \lambda_s; q'))$ where $(\lambda_1, ..., \lambda_s; q')$ meet the conditions inverse to those of formula (11):

1. if $q' \notin \mathcal{H}_j$ then $\lambda_j = r_j$,

2. if $q' \in \mathcal{H}_j$ and $r_j < k_j$ then $\lambda_j = r_j + 1$,

3. if $q' \in \mathcal{H}_j$ and $r_j = k_j$ then $\lambda_j = r_j$.

At the stage $i = n$ the desired p-value is the sum

$$\mathbf{P}(G_n(k_1, ..., k_s)) = \sum_{q \in \mathcal{H}} \mathbf{P}(C_n(k_1, ..., k_s; q)).$$

Markov text model

Tree approach and the recursion (11) can be readily extended to calculate p-values of motif occurrences in random texts generated by the Markov model of order K. Given the order K of the Markov model, the probability $p(a)$ in (11) depends on K previous letters. Thus, if the length $|q|$ of the prefix q is less than K, one cannot calculate $p(a)$ knowing only the prefix q. To overcome this we divide each class $C_i(r_1, ..., r_s; q)$, where $|q| = d < min(K, i)$ into subclasses $C_i(r_1, ..., r_s; q, w)$; each subclass corresponds to a word w of length $min(K, i) - d$. Then, a text T_i of length i belongs to class $C_i(r_1, ..., r_s; q, w)$ if the suffix of T_i of length $min(K, i)$ equals to $w \cdot q$.

Figure 2 gives an example for Markov model of order K = 1. The tree is constructed for the set $\mathcal{H}$ = {AAA, AAC, ACA, ACC, CCT}. The text T = ATGCCAACCTT produces the following sequence of nodes $\{q_i\}_{i \ge 1}$ (the numbers of the corresponding nodes in Figure 2 are shown in square brackets): A[4], (ε, T)[3], (ε, G)[2], C[5], CC[8], A[4], AA[6], AAC[10], ACC[12], CCT[13], (ε, T)[3].

The recursive equations for probabilities $\mathbf{P}(L_n(\mathcal{H}; 1))$, $\mathbf{P}(L_n(\mathcal{H}; k))$, and $\mathbf{P}(L_n(\mathcal{H}_1, ..., \mathcal{H}_s; k_1, ..., k_s))$ can be obtained from the corresponding formulae (7-8), (11–13) and (16) by substituting probabilities $p(a)$ with $p(a|t[1] \cup t[K])$, where

$$t[1] \cdots t[K] = \begin{cases} w \cdot q & \text{if } 0 \le d < K, \\ K\text{-suffix of } q & \text{otherwise.} \end{cases}$$

The Markov extension is currently implemented for K = 1.

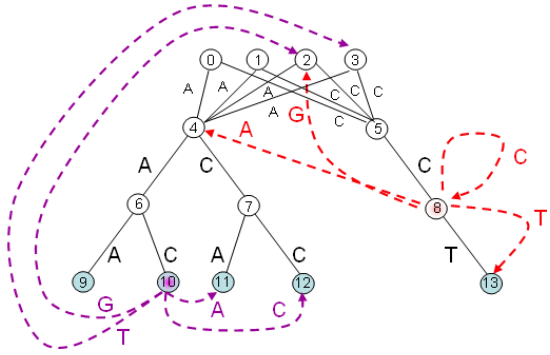

Figure 2

Tree $\mathcal{T}(\mathcal{H})$ for the set $\mathcal{H}$ = {aaa, aac, aca, acc, cct} with dashed links for δ function under Markov(1) model. Tree $\mathcal{T}(\mathcal{H})$ for the set $\mathcal{H}$ = {AAA, AAC, ACA, ACC, CCT} under Markov model of order 1. Dashed colored links represent δ function for internal node (8) – in red, and for marked node (10) corresponding to the word AAC $\in \mathcal{H}$ – in purple.

Complexity

To resume, the computation of $\mathbf{P}(L_n(\mathcal{H}; k))$ for one set $\mathcal{H}$ requires a computation of $\left(\mathbf{P}(C_i(l,q))\right)_{0\le l<k, q\in Q_{\mathcal{H}}}$ for $i \le n$. For each iteration, the time complexity is $O(k|Q_{\mathcal{H}}||\Sigma|)$, where $|\Sigma|$ is the size of the alphabet. One traverses the tree n times. As $|Q_{\mathcal{H}}|$ is upper bounded by $(m|\mathcal{H}|)$, where m is the maximal length of word in $\mathcal{H}$, this yields the overall $O(nkm|\mathcal{H}||\Sigma|)$ time complexity and a $O(km|\mathcal{H}|)$ space complexity.

When several sets are involved, the number of nodes in the tree $\mathcal{T}(\mathcal{H}_1 \cup \cdots \cup \mathcal{H}_s)$ becomes $O(m|\mathcal{H}|)$ with m equal to the maximal length of word in $\mathcal{H}=\mathcal{H}_1 \cup \cdots \cup \mathcal{H}_s$. Additional memory in each node is $\prod_i k_i$. Therefore, the time complexity is $O(nm|\Sigma|\prod_i k_i|\mathcal{H}|)$ and the space complexity is $O(m \prod_i k_i |\mathcal{H}|)$. In the Markov model of order K, one memorizes $|\Sigma|^{K-d}$ predecessors for each node at depth d, $0 = d < K$. In other words, the number of classes becomes $(m|\mathcal{H}| + K|\Sigma|^K)$. Therefore, the space memory is $O((m|\mathcal{H}| + K|\Sigma|^K)\prod_i k_i)$ and the running time is $O(n|\Sigma|(m|\mathcal{H}| + K|\Sigma|^K)\prod_i k_i)$. This *additive* increment compares favorably to simple induction methods [45,53] that introduce a *multiplicative* $O(K|\Sigma|^K)$ factor in time and space complexity for the Markov(K) model.

Results and discussion

We developed an algorithm for precise calculation of the *p*-value for multiple occurrences of multiple motifs with possible overlaps. The running time is linear in the text length and depends on the alphabet size, the maximal motif length, the number of words in the motifs, and the number of occurrences of each motif. The algorithm was implemented in the AHOPRO software. Below we give examples of how *p*-values can be used for studying gene regulation *in silico*, particularly for selecting optimal cutoff values for motifs represented by PWMs. In the subsection '*Comparison with simulation and approximation methods*' we compare our *p*-value computations with the result of Monte Carlo simulations and the Poisson approximation. Our results confirm the accuracy of our algorithm and show in what cases the Poisson approximation [8,11] cannot be employed. In the subsection '*Optimal cutoffs*', we apply AHOPRO to choose an appropriate cutoff score for Position Weights Matrices. In the subsection '*Assessment of gene regulation*', we show how AHOPRO can be used for studying regulatory regions containing heterotypic clusters of TFBSs to distinguish genes that are regulated by given transcription factors from those that are not.

As a model example, we use in this section data published in [34] on regulatory clusters in *D. melanogaster*. This compilation includes information on

(i) known binding motifs for transcription factors,

(ii) known CRM regions, and

(iii) known regulatory interactions.

Comparison with simulation and approximation methods

In our first example we use the *even-skipped stripe 2* enhancer (*eve2*) [63] of length 728 bp that is known to contain binding sites for TFs *bicoid, kruppel* and *hunchback*. Below we compare *p*-values calculated by the AHOPRO program and those calculated using compound Poisson approximation with *p*-values computed through Monte Carlo simulations.

AhoPro and Monte Carlo comparisons

Table 2 displays results of comparison of *p*-values calculated with AHOPRO and with Monte Carlo simulation assuming the Bernoulli model M0. The corresponding results for the first order Markov model M1 are displayed in Table 3. Letters probabilities for M0 and the transition matrix for M1 were evaluated from *eve2* sequence. We used the PWM cutoff values taken from [34], i.e., 5.3, 5.0, and 6.2 for *bicoid, kruppel*, and *hunchback* respectively. With these threshold values in sequence *eve*2 we have

Table 2: Comparison of *p*-values calculated by the AHOPRO program, by Monte Carlo simulations and by compound Poisson distribution formula under the M0 model

MOTIF, CUTOFF	OCC.	AHOPRO	MONTE CARLO	POISSON	AHOPRO/MC	AHOPRO/POISSON
bcd, 5.3	3	0.012	0.012	0.010	1.00	1.10
kr, 5.0	4	0.0044	0.0044	0.0033	1.01	1.34
hb, 6.2	2	0.013	0.013	0.012	0.99	1.04
bcd & *kr*	3&4	0.00025	0.00026	3.6E-05	0.99	7.10
bcd & *kr* & *hb*	3&4&2	6.54E-06	5.8E-06	4.34E-07	1.13	7.13

Comparison of *p*-values calculated for the Markov(0) model by the AHOPRO program with p-values calculated by Monte Carlo simulations and by Poisson formula for motifs of *D. melanogaster* developmental transcription factors *bicoid, kruppel* and *hunchback*.

Table 3: Comparison of *p*-values calculated by the AHOPRO program, by Monte Carlo simulation and by compound Poisson distribution formula under the M1 model

MOTIF, CUTOFF	OCC.	AHOPRO	MONTE CARLO	POISSON	AHOPRO/MC	AHOPRO/POISSON
bcd, 5.3	3	0.013	0.014	0.012	0.998	1.11
kr, 5.0	4	0.011	0.011	0.008	1.01	1.43
hb, 6.2	2	0.14	0.14	0.11	0.9987	1.25
bcd & *kr*	3&4	0.00051	0.00051	9.62E-05	0.9991	5.34
bcd & *kr* & *hb*	3&4&2	6.9E-05	6.97E-05	1.08E-05	0.9889	6.36

Comparison of *p*-values calculated by the AHOPRO program for the Markov(1) model with those calculated by Monte Carlo simulations and by Poisson formula for motifs of *D. melanogaster* developmental transcription factors *bicoid*, *kruppel*, and *hunchback*.

found 3, 4, and 2 occurrences of motifs of each type respectively. In Tables 2 and 3 we listed the *p*-values, i.e, the probabilities to find no less than the observed number of occurrences of motifs in a random text of length L, where L is the length of *eve2* enhancer. The number of Monte Carlo simulations was set to 10^6 everywhere, except for the triplet (*bcd&kr&hb*), where we did 10^7 simulations. The probability to find the observed number of occurrences of (*bcd&kr&hb*) simultaneously in the same simulated sequence is extremely low; thus we increased the number of simulations so that the product of the probability by the number of simulations be greater than 1.

The results of comparison of the AHOPRO computation with those obtained from simulated random sequences presented in Tables 2 and 3 confirm the accuracy of our algorithm.

Poisson approximation

In practical application, compound Poisson distribution [64] is widely used to assess *p*-values of multiple motif occurrences [2,8,34,65]. Here we apply it to compute the probability to observe the given number of motif occurrences when the probabilities of individual words are calculated adopting the M0 or M1 models described above. The results of the comparison given in corresponding columns in Tables 2 and 3 show that the *p*-value calculated using Poisson approximation can be significantly underestimated. This happens most probably because the Poisson approximation does not take into account possible overlaps between motif occurrences and considers motif occurrences as independent. The error increases when the *p*-value is calculated for simultaneous occurrences of several factors, as it is done in the last two rows. In this case, the Poisson approximation *p*-value for a combination of several TFs is calculated as a product of *p*-values calculated independently for each TF. Actually, the motif occurrences can overlap especially when the motifs resemble each other, thus there is no independence, which brings about the error.

Optimal cutoffs

Below, we use AHOPRO to determine the optimal cutoff values for PWMs of regulatory factors, given the sequences of regulatory region assumedly interacting with the factors. The distribution of occurrences of TF binding sites in corresponding experimentally confirmed regulatory regions is strongly biased [34]. In CRMs binding sites often tend to occur in clusters, which is not the case for random sequences.

Different cutoff values correspond to different numbers of putative binding sites of different quality. The higher the cutoff value, the closer the motif occurrences are to the consensus and the smaller the number of motif occurrences. Therefore, for a given factor it is reasonable to select a cutoff value that minimizes the probability of finding in the random sequence the number of motif occurrences observed in the sequence of the regulatory region.

As an example, we considered again transcription factors *bicoid*, *kruppel*, which are known to regulate the *even-skipped stripe 2* (*eve2*) enhancer. To select the optimal cutoff value we used the following procedure: first, in the sequence of *eve2* we counted occurrences of motifs with a score greater than the cutoff with cutoff values varied from 3 to 8.5. Therefore, each pair of cutoff values (S_1, S_2) corresponded to (k_1, k_2) occurrences for motifs of *bicoid* and *kruppel* respectively. For each pair (k_1, k_2), we computed *p*-value $P_n(k_1(S_1), k_2(S_2))$, which is denoted below as $P(S_1, S_2)$. That is the probability to obtain at least k_1 occurrences of *bicoid*, with scores greater than S_1, and at least k_2 occurrences of *kruppel*, with scores greater than S_2. In Figure 3, a 3D-surface is shown, where (x, y, z) corresponds to $(S_1, S_2, -\log_{10} P(S_1, S_2))$, the cutoff value for *bicoid* motif, the cutoff value for *kruppel* motif and -logarithm of the corresponding *p*-value calculated for the M1 model respectively. The view to the surface from the above is shown in Figure 3C. The maximal value for $-\log_{10} P(S_1, S_2)$, 6.3044, is attained when the *bicoid* cutoff is equal to $S_1 = 5.1$ and the *kruppel* cutoff is equal to $S_2 = 5.6$. With such cutoff values in the sequence of the *eve2* enhancer

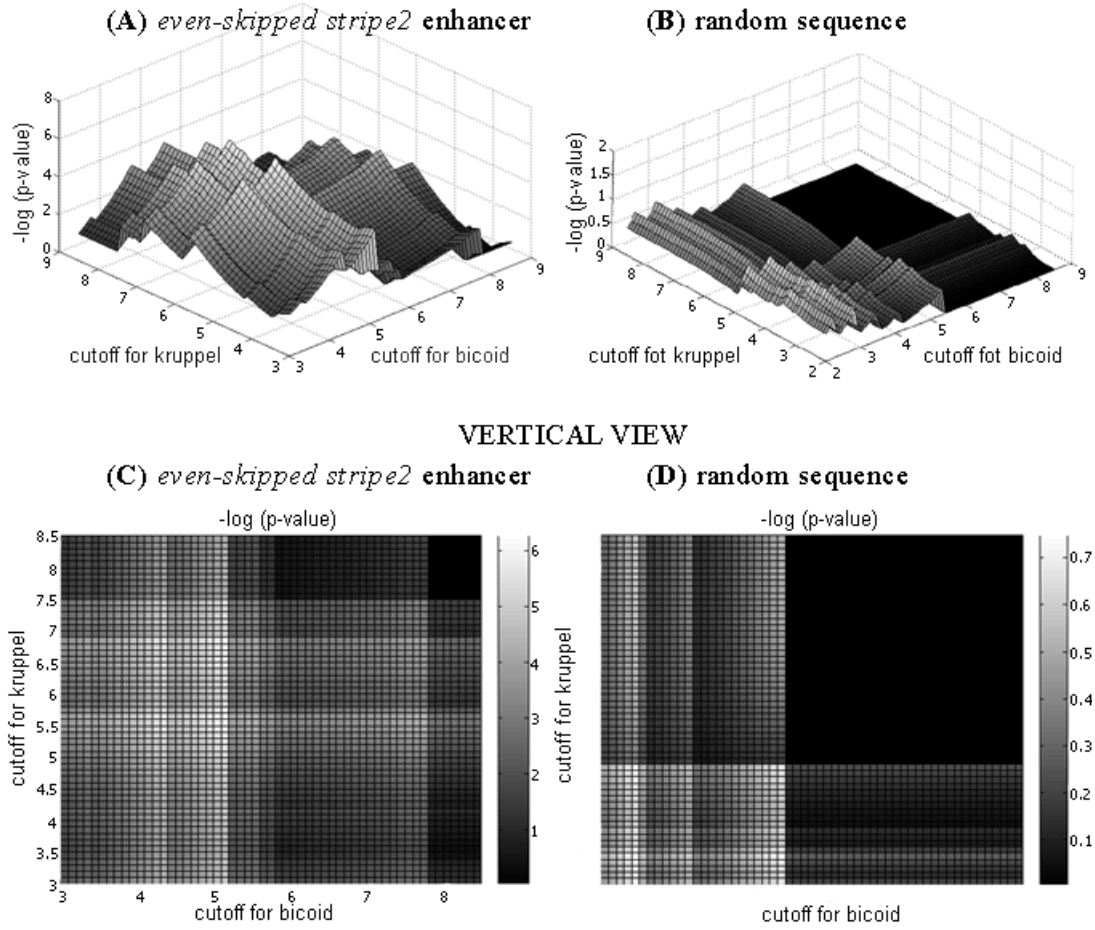

Figure 3

P-value distribution for eve2 and random sequences. Distribution of $\log_{10}$ (*Pvalue*) calculated for the M1 model as a function of cutoff values for PWMs for BICOID and KRUPPEL in the *even-skipped stripe 2* enhancer (A), in a random sequence (B). View from above: eve2 sequence (C), random sequence (D).

there are $k_1 = 6$ and $k_2 = 4$ occurrences of *bicoid* and *kruppel* motifs defined by corresponding PWMs. We believe that the sites that are found with this optimal *p*-value are the best candidates for functional TF binding sites.

For comparison, we simulated random sequences with the same length as the *eve2* enhancer and the same dinucleotide probabilities. In most of simulated sequences, for the cutoff values for *bicoid* and *kruppel* equal to (S_1, S_2) = (5.1, 5.6) we found no more than one occurrence of each motif. The average number of occurrences is 0.54 for *bicoid* and 0.31 for *kruppel*. The average *p*-value is 0.633. We took one of the random sequences and compared *p*-values calculated for various cutoff values in this random sequence (Figures 3B, 3D) and in the real biological sequence of the *eve2* enhancer (Figures 3A, 3C). One can see that there are two major differences between *p*-value distributions in really regulated sequences and in the random sequence. First, *p*-values in the random sequence are much greater than those in the enhancer sequence. In particular, maximal – log(*pvalue*) for this random sequence is about 1.02 which is 6.17 times smaller than maximal – log(*pvalue*) for the enhancer sequence (see also Table 4). Second, the shapes of *p*-value distributions are different. For the enhancer sequence, there are only few distinct peaks (4.3, 5.6),(4.3, 6.8), (5.1, 5.6), (5.1, 6.8) whereas for the random sequence we see ridges between (2.2, 2.0) and (2.2, 4.8), and (2.8, 2.0) and (2.8, 4.8). As we expected, it is impossible to choose the appropriate cutoff for PWMs of factors from the random sequence data (Figures 3B and 3D).

We also would like to address the choice between the M0 and M1 models. We observed, that in almost all cases the

Table 4: Comparison of *p*-values and cutoff for different sets of DNA sequences

regulatory regions bicoid regulated	minimal pvalue	Cut-off	regulatory regions not regulated by bicoid	minimal pvalue	Cut-off	random seq.	minimal pvalue	Cut-off
Btd crm	3.24E-05	3.4	Gt p. enh.	0.023	2.7	seq. 1	0.16	2.6
Hb P2	4.13E-05	3.7	Hb upstream enh.	0.053	4.4	seq. 2	0.12	1.7
Kni cis element	0.01	5.3	Eve stripe 4+6 enh.	0.41	3.6	seq. 3	0.25	1.2
Kr CD-1 enh.	0.0001	5.1	Eve stripe 3+7 enh.	0.58	2.5	seq. 4	0.065	1.6
Otd early enh.	0.024	5	Ftz upstream enh.	0.037	5.8	seq. 5	0.11	1
Sal blastoder. enh.	8.62E-04	6.5	Ftz	0.28	3.3	seq. 6	0.0087	3.8
Tll PD enh.	0.26	4.2	Ubx PBX enh.	0.196	6.7	seq. 7	0.024	2.9
Tll AD+PD enh.	0.025	8.1	Ubx BXD enh.	0.698	4.6	seq. 8	0.17	3.4
Eve stripe 2 enh.	4.04E-05	5.1	Ubx BX enh. (BRE)	0.05	7.5	seq. 9	0.092	2.8
Eve stripe 1 enh.	8.09E-06	5.2	Ems upstream enh.	0.276	4.4	seq. 10	0.052	3.6
Eve stripe 5 enh.	0.27	3.8	En stripe enh. (intr. 1)	0.049	5	seq. 11	0.13	1.7
Median	8.62E-04	5.1	Median	0.196	4.4	Median	0.1128	2.6

Comparison of minimal *p*-values and best found cutoffs for bicoid PWM calculated (i) in regulatory regions which are regulated by *bicoid*, (ii) in regulatory regions which are not regulated by *bicoid*, and (iii) in random sequences of the same length and with the same dinucleotide distribution as in the *even-skipped stripe 2* enhancer.

p-value calculated for the M0 model is smaller than the *p*-value calculated for the M1 model. This can probably be explained by the fact that using the M1 model we take into account more information about the real sequence than in the M0 model. Nevertheless, the difference is not crucial; for instance, the greatest value of the ratio between *p*-values calculated adopting the M0 and M1 for *bicoid* and *kruppel* is about 3.62 for the *eve2* enhancer. So, the M0 model can be equally used in practical applications.

Assessment of gene regulation

Enhancers may contain clusters of TF binding sites for gene regulators. In such cases, *p*-value computation can be used to distinguish genes that are regulated by a given transcription factor from those that are not. To illustrate this, we took PWM for TF *bicoid* and calculated *p*-values for different cutoff values in various sets of sequences:

- regulatory regions which are regulated by *bicoid*, the positive set;

- regulatory regions which are not regulated by *bicoid*, the negative set;

- random sequences of the same length as *eve2* enhancer and with the same dinucleotide distribution, the random set.

Minimal *p*-value and the corresponding cutoff value for 11 sequences in each set are presented in Table 4. Comparing the *p*-values we observed that *p*-values calculated for the positive set generally were significantly smaller than those, calculated for the negative and for the random sets.

The median for the *p*-value in the positive set is equal to 8.62E-04. But there are some exceptions, for instance, the *tailless PD* enhancer with a minimal *p*-value that is equal to 0.26 and the *even-skipped stripe 5* enhancer with the minimal *p*-value that is equal to 0.27. Despite the fact that these genes are reported to be regulated by *bicoid* and that there are experimentally confirmed individual *bicoid* binding sites in these sequences, these sequences do not contain clusters of *bicoid* binding sites.

Most *p*-values calculated for the negative set, (second set in Table 4), are significantly higher than *p*-values calculated for the positive set. But we observed rather small *p*-values for sequences of the *giant posterior* enhancer (0.023), the *hunchback* upstream enhancer (0.053), the *fushi tarazu* upstream enhancer (0.037), the *ultrabithorax BX* enhancer (0.05), and the *engrailed* stripe enhancer (0.049). We believe that this can be explained by the fact that these regions contain clusters of binding sites of regulatory factors with motifs that are similar to the *bicoid* motif. Indeed, it was experimentally shown that TF *kruppel* regulates the *giant* posterior enhancer, TF *tailless* regulates the *hunchback* upstream enhancer and the *ultrabithorax BX* enhancer, and TF *fushi tarazu* regulates the *fushi tarazu* upstream enhancer, the *ultrabithorax BX* enhancer and the *engrailed* stripe enhancer. All these motifs of *kruppel*, *tailless* and *fushi tarazu* exhibit some similarity to the *bicoid* motif. This observation shows the necessity to use some sort of conditional *p*-values in order to distinguish between the true *bicoid* clusters and the clusters of weak *bicoid* sites induced by presence of the clusters of other TF sites [67]. Moreover, the apparent false positive hit (*p*-value = 0.05, cutoff = 7.5) in a region that was not reported to be regulated by *bicoid* seems to be related to

the real *bicoid* binding, although not necessarily functional.

For the random set, i.e., sequences simulated with the same dinucleotide probabilities as in the *even-skipped stripe 2* enhancer, we observe a rather broad range of minimal *p*-values, from 0.0087 for the 6th sample to 0.25 for the 3rd sample. It shows that the predictive power of this approach is limited to the case of regulatory sequences containing clusters of motifs.

Conclusion

In this work we have developed an algorithm inspired by the Aho-Corasick pattern matching algorithm that allows precise calculation of the probability to find given motif conformation in a random text. It was implemented in the AHOPRO software for the Bernoulli model and the Markov model of order 1 of random sequences. There would be no difficulty in extending our approach for Markov models of order k, $k > 1$. We compared probabilities computed with AHOPRO with those computed by compound Poisson distribution and showed that in the case of multiple occurrences of multiple motifs the Poisson approximation often substantially underestimate the *p*-value.

As we have demonstrated, the statistical significance of multiple motif occurrence in the text can be efficiently calculated with a simple algorithm. This can give an independent criteria to improve the results of site extraction algorithms, which still performs rather poorly. P-values or E-values are used in such programs as BLAST and make quantities to which practicing biologists are used to. Thus, adopting this measure to motif extraction (for a single or multiple motif occurrences) would greatly help the users who use motif extraction analysis as a preliminary stage for experiments in the lab. On the other hand, our algorithm is not connected with a particular motif extraction program, and uses a most general motif representation, the list of the allowed words [35], as input. Thus, it can be used when the results of several motif extraction algorithms are compared, for instance in the interpretation of ChIP-chip experiments [5]. In addition, our algorithm AHOPRO can easily be extended to amino acid sequences and applied in identification of protein domain signatures.

Authors' contributions

VM initiated the study by pointing at the biological problem. JC suggested the initial idea of using Aho-Corasick structure. The final version of the algorithm was developed in discussions between JC, VB, MR and MAR. JC and VB developed the implementation. VB obtained results on simulated and biological sequences. VB designed the web site. MR, MAR, VB and VM participated in manuscript writing. MR and VM coordinated the study. All authors read and approved the final manuscript.

Additional material

Additional file 1

Bernoulli text model. Probability to find multiple occurrences of a single motif. The detailed description of the algorithm for the p*-value calculation in the case of multiple occurrences of a single motif.*
Click here for file
[http://www.biomedcentral.com/content/supplementary/1748-7188-2-13-S1.pdf]

Additional file 2

Tree construction from PWM motif representation. The brief description of the procedure of the prefix tree construction from PWM motif representation.
Click here for file
[http://www.biomedcentral.com/content/supplementary/1748-7188-2-13-S2.pdf]

Additional file 3

Tree construction from PWM motif representation. Steps of the prefix tree construction for a PWM and a given cut-off.
Click here for file
[http://www.biomedcentral.com/content/supplementary/1748-7188-2-13-S3.bmp]

Acknowledgements

Thanks to Andrey Mironov, Stephen Small, Dmitri Papatsenko, Bruno Salvy and Philippe Flajolet for helpful comments and suggestions. Thanks to Alexander Favorov for help with the programming. Thanks to Tim Barker for correcting the English in the manuscript. This research was partially supported by INTAS #04-83-3994 and #05-1000008-8028, French Program EcoNet-12635WG, the RFBR grants 07-04-01584 and 06-04-49249, and by Russian Federation Agency in Science and Innovation State Contract 02.531.11.9003.

References

1. MacIsaac KD, Fraenkel E: **Practical strategies for discovering regulatory DNA sequence motifs.** *PloS Comput Biol* 2006, **2(4):**e36.
2. Sandve GK, Drablos F: **A survey of motif discovery methods in an integrated framework.** *Biol Direct* 2006, **1:**11.
3. Rombauts S, Florquin K, Lescot M, Marchal K, Rouze P, van de Peer Y: **Computational approaches to identify promoters and cis-regulatory elements in plant genomes.** *Plant Physiol* 2003, **132(3):**1162-1176. Review.
4. Bulyk ML: **DNA microarray technologies for measuring protein-DNA interactions.** *Curr Opin Biotechnol* 2006, **17(4):**422-30.
5. Harbison CT, Gordon B, Lee TI, Rinaldi NJ, Macisaac KD, Danford T, Hannett NM, Tagne JB, Reynolds DB, Yoo J, Jennings EG, Zeitlinger J, Pokholok DK, Kellis M, Rolfe PA, Takusagawa KT, Lander ES, Gifford DK, Fraenkel E, Young RA: **Transcriptional regulatory code of a eukaryotic genome.** *Nature* 2004, **431:**99-104.
6. Zhu Z, Shendure J, Church GM: **Discovering functional transcription-factor combinations in the human cell cycle.** *Genome Res* 2005, **15(6):**848-55.
7. Clyde DE, Corado MS, Wu X, Pare A, Papatsenko D, Small S: **A self-organizing system of repressor gradients establishes segmental complexity in Drosophila.** *Nature* 2003, **426(6968):**849-53.

8. Wagner A: **Genes regulated cooperatively by one or more transcription factors and their identification in whole eukaryotic genomes.** *Bioinformatics* 1999, **15(10):**776-784.
9. Lifanov AP, Makeev VJ, Nazina AG, Papatsenko DA: **Homotypic regulatory clusters in Drosophila.** *Genome Res* 2003, **13(4):**579-88.
10. Brown CT, Rust AG, Clarke PJ, Pan Z, Schilstra MJ, De Buysscher T, Griffin G, Wold BJ, Cameron RA, Davidson EH, Bolouri H: **New computational approaches for analysis of cis-regulatory networks.** *Dev Biol* 2002, **246:**86-102.
11. Wagner A: **A computational genomics approach to the identification of gene networks.** *Nucleic Acids Res* 1997, **25(18):**3594-3604.
12. Liaw GJ, Lengyel JA: **Control of tailless expression by bicoid, dorsal and synergistically interacting terminal system regulatory elements.** *Mech Dev* 1993, **40(1–2):**47-61.
13. Jun S, Desplan C: **Cooperative interactions between paired domain and homeodomain.** *Development* 1996, **122(9):**2639-50.
14. Mitashev VI, Koussoulakos S, Zinov'eva RD, Ozerniuk ND, Mikaelian AS, Shmukler E, Smirnova Iu A: **[Constructive synergism of regulatory genes expressed in the course of the eye and muscle development and regeneration].** *Izv Akad Nauk Ser Biol* 2001:261-75.
15. Klingenhoff A, Frech K, Werner T: **Regulatory modules shared within gene classes as well as across gene classes can be detected by the same in silico approach.** In *Silico Biol* 2002, **2:**S17-26.
16. Kato M, Hata N, Banerjee N, Futcher B, Zhang MQ: **Identifying combinatorial regulation of transcription factors and binding motifs.** *Genome Biol* 2004, **5(8):**R56. Epub 2004 Jul 28.
17. Hu YJ, Sandmeyer S, McLaughlin C, Kibler D: **Combinatorial motif analysis and hypothesis generation on a genomic scale.** *Bioinformatics* 2000, **16(3):**222-32.
18. Jegga AG, Sherwood SP, Carman JW, Pinski AT, Phillips JL, Pestian JP, Aronow BJ: **Detection and visualization of compositionally similar cis-regulatory element clusters in orthologous and coordinately controlled genes.** *Genome Res* 2002, **12(9):**1408-17.
19. Li H, Rhodius V, Gross C, Siggia ED: **Identification of the binding sites of regulatory proteins in bacterial genomes.** *Proc Natl Acad Sci USA* 2002, **99(18):**11772-7. Epub 2002 Aug 14.
20. Markstein M, Zinzen R, Markstein P, Yee KP, Erives A, Stathopoulos A, Levine M: **A regulatory code for neurogenic gene expression in the Drosophila embryo.** *Development* 2004, **131(10):**2387-94.
21. Makeev V, Lifanov A, Nazina A, Papatsenko D: **Distance preferences in distribution of binding motifs and hierarchical levels in organization of transcription regulatory information.** *Nucleic Acids Res* 2003, **31(20):**6016-26.
22. Halfon MS, Michelson AM: **Exploring genetic regulatory networks in metazoan development: methods and models.** *Physiol Genomics* 2002, **10(3):**131-43.
23. Papatsenko D: **ClusterDraw web server: a tool to identify and visualize clusters of binding motifs for transcription factors.** *Bioinformatics* 2007, **23(8):**1032-1034.
24. Aerts S, Loo PV, Thijs G, Moreau Y, Moor BD: **Computational detection of cis -regulatory modules.** *Bioinformatics* 2003, **19(2):**II5-II14.
25. Bailey T, Noble W: **Searching for statistically significant regulatory modules.** *Bioinformatics* 2003, **19(2):**II16-II25.
26. Berman B, Pfeiffer B, Laverty T, Salzberg S, Rubin G, Eisen M, Celniker S: **Computational identification of developmental enhancers: conservation and function of transcription factor binding-site clusters in Drosophila melanogaster and Drosophila pseudoobscura.** *Genome Biol* 2004, **5(9):**R61.
27. Frith M, Hansen U, Weng Z: **Detection of cis-element clusters in higher eukaryotic DNA.** *Bioinformatics* 2001, **17(10):**878-889.
28. Frith MC, Li MC, Weng Z: **Cluster-Buster: Finding dense clusters of motifs in DNA sequences.** *Nucleic Acids Res* 2003, **31(13):**3666-3668.
29. Sosinsky A, Bonin C, Mann R, Honig B: **Target Explorer: an automated tool for the identification of new target genes for a specified set of transcription factors.** *Nucleic Acids Research* 2003, **31(13):**3589-3592.
30. Krivan W: **Searching for transcription factor binding site clusters: how true are true positives?** *J Bioinform Comput Biol* 2004, **2(2):**413-6.
31. Papatsenko D, Makeev V, Lifanov A, Régnier M, Nazina A, Desplan C: **Extraction of Functional Binding Sites from Unique Regulatory Regions: The *Drosophila* Early Developmental Enhancers.** *Genome Research* 2002, **12:**470-481. [Preliminary version in Drosophila Workshop, Washington 2001].
32. Markstein M, Markstein P, Markstein V, Levine M: **Genome-wide Analysis of Clustered Dorsal Binding Sites Identifies Putative Target Genes in the Drosophila Embryo.** *PNAS* 2002, **99(2):**763-768.
33. Rebeiz M, Reeves NL, Posakony JW: **SCORE: a computational approach to the identification of cis-regulatory modules and target genes in whole-genome sequence data. Site clustering over random expectation.** *Proc Natl Acad Sci USA* 2002, **99(15):**9888-93. Epub 2002 Jul 09.
34. Lifanov A, Makeev V, Nazina A, Papatsenko D: **Uniform clusters in Drosophila.** *Genome Res* 2003, **13(4):**579-588.
35. Staden R: **Methods for calculating the probabilities of finding patterns in sequences.** *Comput Appl Biosci* 1989, **5(2):**89-96.
36. Ellington A, Szostak J: **In vitro selection of RNA molecules that bind specific ligands.** *Nature* 1990, **346:**818-822.
37. Tuerk C, Gold L: **Systematic evolution of ligands by exponential enrichment: RNA ligands to bacteriophage T4 DNA polymerase.** *Science* 1990, **249:**505-510.
38. Berger MF, Philippakis AA, Qureshi AM, He FS, Estep PW, Bulyk ML: **Compact, universal DNA microarrays to comprehensively determine transcription-factor binding site specificities.** *Nat Biotechnol* 2006, **24:**1429-1435.
39. Liu Y, Yokota H: **Modeling Transcriptional Regulation in Chondrogenesis Using Particle Swarm Optimization.** *IEEE Symposium on Computational Intelligence in Bioinformatics and Computational Biology, CIBCB2005* 2005:311-317.
40. **IUPAC codes** [http://bioinformatics.org/sms2/iupac.html]
41. Berg OG: **Selection of DNA binding sites by regulatory proteins. Functional specificity and pseudosite competition.** *J Biomol Struct Dyn* 1988, **6(2):**275-297.
42. Knuth DE: *The Art of Computer Programming, Sorting and Searching Volume 3*. Addison-Wesley; 1973.
43. Zhang J, Jiang B, Li M, Tromp J, Zhang X, Zhang M: **Computing exact P-values for DNA motifs.** *Bioinformatics* 2007, **23(5):**531-537.
44. Hertzberg L, Zuk O, Getz G, Domany E: **Finding Motifs in Promoter Regions.** *Journal of Computational Biology* 2005, **12(3):**314-330.
45. Robin S, Daudin JJ: **Exact distribution of word occurrences in a random sequence of letters.** *J Appl Prob* 1999, **36:**179-193.
46. Chrysaphinou C, Papastavridis S: **The Occurrence of Sequence of Patterns in Repeated Dependent Experiments.** *Theory of Probability and Applications* 1990, **79:**167-173.
47. Guibas L, Odlyzko A: **String Overlaps, Pattern Matching and Nontransitive Games.** *Journal of Combinatorial Theory, Series A* 1981, **30:**183-208.
48. Tanushev M, Arratia R: **Central Limit Theorem for Renewal Theory for Several Patterns.** *Journal of Computational Biology* 1997, **4:**35-44.
49. Nicodème P, Salvy B, Flajolet P: **Motif Statistics.** *Theoretical Computer Science* 2002, **287(2):**593-618. [Preliminary version at ESA'99].
50. Régnier M: **A Unified Approach to Word Occurrences Probabilities.** *Discrete Applied Mathematics* 2000, **104:**259-280. [Special issue on Computational Biology;preliminary version at RECOMB'98].
51. Szpankowski W: *Average Case Analysis of Algorithms on Sequences* New York: John Wiley and Sons; 2001.
52. Bassino F, Clément J, Fayolle J, Nicodème P: **Counting occurrences for a finite set of words: an inclusion-exclusion approach.** *2007 International Conference on Analysis of Algorithms (AofA'07), Discrete Mathematics and Theoretical Computer Science* 2007:12.
53. Park Y, Spouge J: **Searching for Multiple Words inMarkov Sequences.** *INFORMS journal of Computing* 2004, **16(4):**341-347.
54. Nicodème P: **Regexpcount, a symbolic package for counting problems on regular expressions and words.** *Fundamenta Informaticae* 2003, **56(1–2):**71-88.
55. Klaerr-Blanchard M, Chiapello H, Coward E: **Detecting localized repeats in genomic sequences: A new strategy and its appli-**

cation to *B. subtilis* and *A. thaliana* sequences. *Comput Chem* 2000, **24:**57-70.
56. Reinert G, Schbath S: **Compound Poisson Approximation for Occurrences of Multiple Words in Markov Chains.** *Journal of Computational Biology* 1998, **5(2):**223-253.
57. Régnier M, Vandenbogaert M: **Comparison of statistical significance criteria.** *J Bioinform Comput Biol* 2006, **4(2):**537-551.
58. Régnier M: **Mathematical Tools for Regulatory Signals Extraction.** In *Bioinformatics of Genome Regulation and Structure* Edited by: Kolchanov N, Hofestaedt R. Kluwer Academic Publisher; 2004:61-70. [Preliminary version at BGRS'02].
59. Régnier M, Denise A: **Rare events and Conditional Events on random strings.** *DMTCS* 2004, **6(2):**191-214.
60. Boeva V, Clément J, Régnier M, Vandenbogaert M: **Assessing the significance of Sets of Words.** In *CPM'05, of Lecture Notes in Computer Science Volume 3537*. Springer-Verlag; 2005:358-370. [Proc. CPM'05, Jeju Island, Korea].
61. Kucherov G, Noé L, Roytberg M: **Multi-seed lossless filtration.** In *Proceedings of the 15th Annual Combinatorial Pattern Matching Symposium (CPM), Istanbul (Turkey), of Lecture Notes in Computer Science Volume 3109*. Edited by: Sahinalp S, Muthukrishnan S, Dogrusoz U. Springer Verlag; 2004:297-310.
62. Aho A, Corasick M: **Efficient String Matching.** *CACM* 1975, **18(6):**333-340.
63. Small S, Blair A, Levine M: **Regulation of even-skipped stripe 2 in the Drosophila embryo.** *Embo Journal* 1992, **11(13):**4047-4057.
64. Reinert G, Schbath S: **Compound Poisson and Poisson process approximations for occurrences of multiple words in Markov chains.** *J Comput Biol* 1998, **5(2):**223-53.
65. Wasserman W, Fickett J: **Identification of regulatory regions which confer muscle-specific gene expression.** *J Mol Biol* 1998, **278:**167-81.
66. Tompa M, Li N, Bailey T, Church G, De Moor B, Eskin E, Favorov A, Frith M, Fu Y, Kent J, Makeev V, Mironov A, Noble W, Pavesi G, Pesole G, Régnier M, Simonis N, Sinha S, Thijs G, van Helden J, Vandenbogaert M, Weng Z, Workman C, Ye C, Zhu Z: **An Assessment of Computational Tools for the Discovery of Transcription Factor Binding Sites.** *Nature Biotechnology* 2005, **23:**137-144.
67. Blanchette M, Sinha S: **Separating real motifs from their artifacts.** *Bioinformatics* 2001, **17(Suppl 1):**S30-8.