Fast Approximate Motif Statistics *

Pierre Nicodème
DKFZ Theoretische Bioinformatik
Im Neuenheimer Feld 280
69120 Heidelberg
Germany

July 3, 2001

Abstract

We present in this article a fast approximate method for computing the statistics of number of non self-overlapping matches of motifs in a random text in the non-uniform Bernoulli model. This method is well suited for protein motifs where the probability of self-overlap of motifs is small. For 96% of the Prosite motifs, the expectations of occurrences of the motifs in a 7 million amino-acids random database are computed by the approximate method with less than 1% error when compared with the exact method. Processing of the whole Prosite takes about 30 seconds with the approximate method. We apply this new method to a comparison of the *C. elegans* and *S. cerevisiae* proteoms.

1 Introduction

Motifs appear in molecular biology as signatures for families of similar sequences. Prosite motifs are a subclass of regular expressions. They do not contain the Kleene star-operator $\star$ defining indefinite repetition and generate therefore finite languages.

Most notable research about statistics on words has been done by Pevzner *et al.* [7], Schbath *et al.* [13], Prum *et al.* [8], Reinert and Schbath [11], Sewell and Durbin [15], Atteson [1], and Régnier and Szpankowski [9, 10]. We consider matches with infinite languages that are restricted to be non self-overlapping. These languages therefore both are more restricted and more general than those considered by Reinert and Schbath [11] and Régnier and Szpankowski [9, 10]. These authors consider matches with finite set of words, a theoretical framework to which belong matches with Prosite motifs. However their methods imply a complexity that is quadratic in the size of the set of words considered; this is practically intractable for most Prosite motifs that define languages of large cardinality.

Nicodème *et al.* [6] proposed an exact method for computing the statistics of occurrences of regular expressions (or motifs) in random texts generated by Bernoulli or Markov sources. They construct a marked automaton for the input regular expression, use the Chomski-Schützenberger algorithm to produce the system of linear equations corresponding to the automaton and solve this system to get the bivariate generating function counting the expected number of matches. The complexity is not function of the cardinality of the language, but function of the size of the automaton. The worst case has exponential complexity in the size of the motif. On the average, the exact computation takes 4 minutes for one Prosite motif; however 10% of the motifs have too high computational complexity and cannot be computed.

We present here a simple and fast approximate method to compute in constant time the statistics of the number of matches of motifs in random texts without constructing an automaton. The results are exact for non self-overlapping motifs and approximate in the other cases. We consider the non uniform Bernoulli case, while the exact method developed in [6] coped with the more general Markov model. For

*Present address: Laboratoire Statistiques et Génomes, CNRS - Génopole Evry, 523 Place des Terrasses, 91000 - Evry, France. Email: `Pierre.Nicodeme@inria.fr`.

proteins motifs, the comparison of the approximate and exact methods is good, corresponding to small probability of self-overlap of the motifs, and the Bernoulli model is sufficient.

We prove that, when the number of occurrences in large texts is small, the limiting distribution is Poisson, and use the tail distribution of the Poisson to calibrate the number of matches observed. This gives a quality measure of the over or under-representation of a motif in a text.

As an application, we obtain the asymptotic statistics of the number of matches of the PROSITE motifs in the *C. elegans* and the *S. cerevisiae* proteoms.

This article is organized as follows. In Section 2 we review the basic results about generating functions and languages. We describe in Section 3 the fast approximate method. We prove in Section 4 the Poisson law for rare occurrences. Section 5 validates the method by comparing exact and approximate statistics of occurrences of PROSITE motifs in the ProDom database. In Section 6 we apply the method to calibrate the occurrences of the PROSITE motifs in *C. elegans*. and *S. cerevisiae*, which makes the exceptional motifs conspicuous. Section 7 deals with implementation and performances.

2 Languages and generating functions

A textbook of reference for this is Flajolet and Sedgewick [14]. Other references are [12] and [4].

2.1 Definitions

We consider a *finite alphabet* $\Sigma = \{\ell_1, \dots, \ell_p\}$. A *word* or *text* is a finite sequence of *letters* (elements) of Σ. A *language* over Σ is a set of words. The *product language* $A = A_1 \cdot A_2$ is the set of words obtained by concatenation of a word of A_1 and a word of A_2. This definition extends immediately to products of more than two languages. The *star closure* $A^\star$ of a language A is the infinite union $\bigcup_{k\geq 0} A^k$, where A^0 is the empty language containing the empty word (noted ϵ). The language $\Sigma^\star$ is the collection of all possible words over Σ.

The product $B = A_1 \cdot A_2$ of two languages A_1 and A_2 is *ambiguous* if there exists a word w in B such that $w = w_1.w_2 = w_1'.w_2'$ where w_1 and $w_1' \in A_1$, w_2 and $w_2' \in A_2$ and $w_1 \neq w_1'$. In this case, ambiguity implies that there are two or more possible decompositions of the word w over A_1 and A_2. This generalizes to products of more than two languages.

A text $t_1 t_2 \dots t_n$ *matches* a language L at position k if there is at least one word $w \in L$ such that $t_1 \dots t_k = uw$.

We consider the number of matches of a regular expression or a language in a text in the left to right *non-overlapping* or *renewal* context, where the text is scanned from left to right, and every time a match is found, the count is incremented and the search starts afresh at this position.

2.2 Regular languages and motifs

The *regular languages* over the alphabet $\Sigma = \{\ell_1, \dots, \ell_p\}$ are the set of languages recursively built by a finite number of unions, products and star closure over the set of languages $\{\{\epsilon\}, \{\ell_1\}, \dots, \{\ell_p\}\}$. Regular expression are short-cut descriptions of regular languages, where ℓ denotes the singleton language $\{\ell\}$ and $+$ often denotes union. The order of precedence for the operators is $\star, \cdot, +$.

2.3 Generating functions of a language

Multivariate generating function counted by letters

The multivariate generating function $A(\ell_1, \dots, \ell_p)$ of a language A counted by letters is

$$A(\ell_1, \dots, \ell_p) = \sum_{w\in A} \ell_1^{\ell_1(w)} \dots \ell_p^{\ell_p(w)} = \sum a_{i_1,\dots,i_p} \ell_1^{i_1} \dots \ell_p^{i_p}, \tag{1}$$

where $\ell_k(w)$ is the number of letters of w equal to ℓ_k and $a_{i_1,\dots,i_p}$ counts the number of words of the language with i_k letters equal to ℓ_k for k from 1 to p. In the notation $A(\ell_1, \dots, \ell_p)$, the letters ℓ_k are symbolic variables of the expression.

Informally, the multivariate generating function of the language is obtained by adding all the words of A and allowing the letters to commute.

Probability generating function

In the Bernoulli case, where the letter ℓ_k has probability π_k, the probability generating function $A_\pi(z)$ of the language A is obtained from the multivariate generating function counted by letters by the symbolic substitution $\ell_k \to \pi_k z$ for k from 1 to p. This gives

$$A_\pi(z) = A(\pi_1 z, \ldots, \pi_p z) = \sum_{w \in A} \pi_1^{\ell_1(w)} z^{\ell_1(w)} \ldots \pi_p^{\ell_p(w)} z^{\ell_p(w)} = \sum_{w \in A} \pi_1^{\ell_1(w)} \ldots \pi_p^{\ell_p(w)} z^{|w|} = \sum_{w \in A} \pi(w) z^{|w|}.$$

where $\pi(w)$ is the probability of occurrence of the word w at a random position of the text.

Bivariate generating function for the number of matches

Let $p_{n,k}$ be the probability that a random text of size n contains exactly k matches with A. The bivariate generating function counting the matches is

$$A_\pi(z,u) = \sum_{n,k \geq 0} p_{n,k} u^k z^n.$$

This bivariate generating function gives immediate access to the generating functions of the moments of the statistics by differentiation in u. We have

$$E(z) = \sum e_n z^n = \left.\frac{\partial A_\pi(z,u)}{\partial u}\right|_{u=1}, \quad \text{and} \quad M^{(2)}(z) = \sum m_n^{(2)} z^n = \left.\frac{\partial}{\partial u} u \frac{\partial A_\pi(z,u)}{\partial u}\right|_{u=1}, \tag{2}$$

where e_n and $m_n^{(2)}$ respectively are the expectation and the second moment of the number of matches in random texts of size n.

See [6] for algorithms to compute the bivariate function of number of matches $A_\pi(z,u)$ for a regular expression A and exact or asymptotic methods to extract the expectation and variance of the statistics in texts of size n.

Translation from language operators to operations on generating functions

Under the conditions of non-ambiguity of products and disjointness of unions, Union, Product and Star-Closure operations on languages translate to Sum, Product and Quasi-Inverse operations on the corresponding (univariate or multivariate) generating functions. (See Salomaa and Soittola [12]).

3 Fast approximate method

We detail in this section the case of non self-overlapping languages (see the definition below) where neither the construction of an automaton nor solving a large system of linear equations are necessary to get the statistics of number of matches.

For motifs that are finite set of words, Régnier and Szpankowski [9, 10] provide explicit formulas for the expectation and the variance of the number of matches. In this case, setting to zero the correlation functions between the words give the formulas we obtain in the non-overlapping case. It is possible to extend Régnier and Szpankowski results to the infinite non-overlapping case. However, considering infinite languages (or languages with high cardinality like most PROSITE motifs) induces infinite summations, or summations over a very large set. We give here for this case a direct proof which avoids infinite summations, and we compute the expectation of the number of matches in linear time with respect of the number of symbols contained in the motifs.

We define now the languages that we consider.

Definition 1. Two words w_1 and w_2 are overlapping if a prefix of one of these words is a suffix of the other word.

A language $\mathcal{L}$ is self-overlapping if there exist words $w_1, w_2 \in \mathcal{L}$ that overlap.

A language is nested if it contains two words w_1 and w_2 such that w_2 is a factor (substring) of w_1. We include in the set of nested languages the degenerate (in the sense of pattern-matching) languages containing the empty word ϵ.

A regular expression is self-overlapping (*resp.* nested) if the corresponding regular language is self-overlapping (*resp.* nested).

The following example shows that the non-ambiguity condition for products of languages requires some care.

Example 1. Consider languages A, B, and C, such that $A = B = C = \{a, b, ab\}$. Then, AB and BC are non-ambiguous products, but ABC is an ambiguous product ($abab \in ABC = a{\cdot}b{\cdot}ab = ab{\cdot}a{\cdot}b$).

The following lemma proves the unambiguity of concatenation of a non-overlapping language L and of the language N containing no word of L.

Lemma 1. *Let L be a non self-overlapping and non nested language and let $N = \Sigma^\star - \Sigma^\star L \Sigma^\star$ be the language of words with no match with L. The products of languages $N(LN)^i, \forall i \geq 1$, are non-ambiguous.*

Proof. We prove that ambiguous decompositions would lead to a contradiction.

Let us suppose that $w = n_1 \cdot r_1 \cdot n_2 \cdots = n_1' \cdot r_1' \cdot n_2' \cdots \in N(LN)^i$ are two different decompositions of the word w, with $n_1, n_2, \ldots, n_1', n_2', \cdots \in N$ and $r_1, \ldots, r_1' \cdots \in L$.

We suppose that $|n_1| > |n_1'|$ ($|n_1'| > |n_1|$ is the symmetrical case). We therefore have $n_1 = n_1' u$ with $|u| > 0$. If $|n_1| \geq |n_1' r_1'|$ we have $n_1 = n_1' r_1' u'$ with $|u'| \geq 0$. This implies $n_1' r_1' u' \in N$, which since $r_1' \in L$ contradicts the hypothesis that N has no match with L. Therefore $|n_1| < |n_1' r_1'|$.

If $|n_1 r_1| \leq |n_1' r_1'|$ we have $n_1' r_1' = n_1 r_1 v = n_1' u r_1 v$ with $|v| \geq 0$. Therefore $r_1' = u r_1 v$ with $|u| > 0$ which contradicts the hypothesis that L is not nested.

We are left with the case $|n_1 r_1| > |n_1' r_1'|$. We have $n_1 r_1 = n_1' r_1' v' = n_1' u r_1$. This gives $r_1' v' = u r_1$. Moreover $|n_1| = |n_1' u| = |n_1'| + |u| < |n_1' r_1'| = |n_1'| + |r_1'|$, which implies that $|r_1'| > |u|$ and therefore $|r_1| > |v'|$. We get then $r_1' = u\rho_2$ and $r_1 = \rho_1 v'$, which gives $u\rho_2 v' = u \rho_1 v'$ and $\rho_1 = \rho_2$; the words r_1 and r_1' overlap, which contradicts the hypothesis that L is not self-overlapping. This implies that $n_1 = n_1'$.

The equality $n_1 = n_1'$ implies that $|n_1| = |n_1'|$. Then, if $r_1 \neq r_1'$, either r_1' is a factor of r_1 or r_1 is a factor of r_1', which contradicts the hypothesis that L is not nested. Therefore, $r_1 = r_1'$.

If $i = 1$, we obtain $n_2 = n_2'$, and the product NRN is unambiguous. A recurrence on i proves for all i equalities of r_i and r_i', and of n_{i+1} and n_{i+1}'.

Any word $w \in N(LN)^i$ has therefore only one decomposition as concatenation of words of L and N. The products $N(LN)^i$ are unambiguous. □

Remark: this lemma applies to languages beyond the class of regular languages. In particular, the context-sensitive language $L = \bigcup_{n>1} a^n b^n c^n$ verifies the lemma.

3.1 Generating functions for the number of matches

Theorem 1. *Let L be a non self-overlapping and non nested language with probability generating function $L(z)$. Let $F(z, u)$ be the bivariate probability generating function counting the number of matches of L in a random text in the left to right renewal context. Then, we have*

$$F(z, u) = \frac{1}{1 - z + (1 - u)L(z)}. \tag{3}$$

Proof. The demonstration of the theorem follows from marking the texts of $\Sigma^\star$ and by parsing the texts and the marked texts with the occurrences of the motif.

As a consequence of Lemma 1 and with the notations of this lemma, we can decompose $\Sigma^\star$ unambiguously as

$$\Sigma^\star = (NL)^\star N = \bigcup_{i \geq 0} (NL)^i N. \tag{4}$$

In the right member of Equation 4 each text occurs exactly once according to the number of occurrences of the motif.

We consider now a text $t \in \Sigma^\star$ and we re-write it as a marked text t_u ($u \notin \Sigma$) where the mark u has been written after each occurrence of a word of L (in the left to right renewal context). We consider now the mapping $t \to \phi_t(z, u) = \pi(t) u^k z^{|t|}$, where $\pi(t)$ is the probability of the text t and k is the number of occurrences of words of L in t.

Summing up $\phi_t(z, u)$ over all possible texts t gives

$$F(z, u) = \sum_{t \in \Sigma^\star} \phi_t(z, u) = \sum_{t \in \Sigma^\star} \pi(t) u^k z^{|t|} = \sum p_{n,k} u^k z^n,$$

where $F(z,u)$ is the probability generating function of the language Σ_u and $p_{n,k}$ is the probability that a random text of size n contains k occurrences of words of L.

Now, the set of marked texts $\Sigma_u^\star$ may also be rewritten by parsing with the occurrences of the words of L

$$\Sigma_u^\star = (NLu)^* N = \bigcup_{t\in\Sigma^\star} t_u. \tag{5}$$

Each possible text t_u is decomposed in this way. We note that the products $(NLu)^i$ are non-ambiguous and that the intersection of $(NLu)^i$ and $(NLu)^j$ is empty if i and j are different.

Equation 5 immediately translates to the following equation on generating functions

$$F(z,u) = \frac{N(z)}{1 - uN(z)L(z)}, \tag{6}$$

where $N(z)$ is the probability generating function of the language N. But we have $F(z,1) = 1/(1-z)$, this probability generating functions counting all the texts of $\Sigma_u^\star$ where the mark u has been erased, or equivalently, all the texts of $\Sigma^\star$. This gives the equation

$$\frac{1}{1-z} = \frac{N(z)}{1 - N(z)L(z)}. \tag{7}$$

Eliminating $N(z)$ between Equation 6 and Equation 7 provides Equation 3. □

Example 2. We consider the regular expression $R_1 = ba^+c$ over the alphabet $\Sigma = \{a,b,c\}$ with letters probability $\{\pi_a, \pi_b, \pi_c\}$. It is an unambiguous, non self-overlapping and non-nested regular expression. The probability generating function of the matches of R_1 in random texts is

$$F(z,u) = \frac{1}{1 - z + (1-u)\frac{\pi_a\pi_b\pi_c z^3}{1-\pi_a z}}.$$

Example 3. We consider now the regular expression $R_2 = R_1 + bR_1c$. This regular expression is not self-overlapping but it is nested. The words b, c, bb and cc belong to $N_2 = \Sigma^\star - \Sigma^\star R_2 \Sigma^\star$, the language of words with no match with R_2. The words bac and $bbacc$ belong to $R_2 = ba^+c + bba^+cc$. The word $bbbaccc$ may be parsed in two different ways by $N_2R_2N_2$. This specification is ambiguous, as stated by Lemma 1. The automata construction detects self-overlapping or nested constructions and provides for R_2 the same multivariate generating function as for R_1 alone. On the contrary, the multivariate generating function computed by Theorem 1 is

$$F(z,u) = \frac{1}{1 - z + (1-u)\frac{\pi_a\pi_b\pi_c z^3(1+\pi_b\pi_c z^2)}{1-\pi_a z}},$$

in which an extra and erroneous factor $(1 + \pi_b\pi_c z^2)$ appears in the denominator.

3.2 Asymptotic evaluations

We first prove that if a language L is non nested, its probability generating function $L(z)$ is analytic in a disk of radius strictly greater than 1.

Lemma 2 applies to prefix-free languages that we define.

Definition 2. A language L is prefix-free if no word of L is prefix of another word of L.

Remark that non nested languages are prefix-free.

Lemma 2. *The probability generating function $L(z)$ of a prefix-free language L is analytic inside a disk of radius τ strictly greater than 1.*

Proof. Let $L_k = \{w \in L, |w| = k\}$ be the set of words of L of size k. Then, L being a prefix free language, $\forall j > k$, we have $L_j \subset \Sigma^j - L_k\Sigma^{j-k}$. This implies that the sets $M_j = L_j\Sigma^{i-j}$, $j < i$ are disjoint, this last property being true for all i. Moreover $\bigcup_{1\leq j\leq i} M_j \subset \Sigma^i$.

Let $L_k(z)$, $M_k(z)$ and $\Sigma^k(z)$ be the probability generating functions of L_k, M_k and Σ^k, respectively. We have $\Sigma^k(1) = 1$; $M_j(1)$ is the sum of the probability of the words $w \in M_j$ and $M_j(1) = L_j(1)$. This gives

$$\sum_{1\leq j\leq i} M_j(1) \leq 1, \quad \forall i,$$

which implies that $L(1) = \sum_{i\geq 1} L_i(1) \leq 1$ is bounded. The probability generating function $L(z) = \sum l_n z^n$ is expanded as a series with positive coefficients. Therefore, the radius τ of the disc of convergence of the series $L(z)$ is strictly greater than 1 (see Titchmarsh [17]). □

Application of this result and use of standard singularity analysis provide asymptotic approximations for the expectation and the variance of the number of matches.

Theorem 2. *Let L be a non self-overlapping and non nested language with probability generating function $L(z)$. The expectation μ and the variance σ^2 of the number of matches of L with a random text of size n are asymptotically*

$$\mu = L(1) \times n + L(1) - L'(1) + O(A^n), \text{ and} \tag{8}$$

$$\begin{aligned}\sigma^2 = \quad & \{L(1) + L(1)^2 - 2L(1)L'(1)\} \times n \\ & - 2L(1)L'(1) + L(1) + 2L(1)L"(1) + L(1)^2 - L'(1) + L'(1)^2 + O(A^n),\end{aligned} \tag{9}$$

with $A < 1$.

Proof. Lemma 2 states that the radius of convergence ρ of $L(z)$ is strictly greater than 1. Therefore, there exists a positive number A, such that $1 < 1/A < \rho$ and $L(z)$ is analytic in the disc of radius $1/A$ so as its derivatives. The theorem follows then directly of the Taylor expansion

$$L(z) = L(1) + (z-1)L'(1) + \frac{1}{2}(z-1)^2 L"(1) + o((z-1)^2),$$

and by application of the formula

$$[z^n]F(z) = \frac{1}{2i\pi}\oint_\Gamma \frac{1}{z^{n+1}}F(z),$$

where $[z^n]F(z)$ is the n^{th} Taylor coefficient of $F(z)$, the Cauchy contour is a circle centered at the origin and of radius $1/A$, and $F(z)$ is successively taken as

$$E(z) = \frac{L(z)}{(1-z)^2}, \quad \text{and} \quad M^{(2)}(z) = \frac{L(z)}{(1-z)^2} + 2\frac{L(z)^2}{(1-z)^3},$$

respectively the generating functions of the two first moments of the statistics of the number of matches, computed by Equation 2. □

3.3 Non self-overlapping and non nested regular languages

Theorems 1 and 2 apply to non self-overlapping and non nested regular languages. For a non-ambiguous regular expression R, a recursive translation from operators over languages to operators over generating functions provides the rational generating function $R(z)$ of the corresponding regular language. The computation of $R'(z)$ follows by formal differentiation. Both computations are done in time complexity linear with respect to the size of the regular expression. Moreover, when we consider motifs that generate finite languages, $R(z)$ is a polynomial.

This gives the following lemma.

Lemma 3. *If R is a non ambiguous, non self-overlapping and non nested regular expression, the expectation μ and the variance σ^2 of the number of matches of R with a random text of size n are asymptotically given by Formulas 8 and 9. The values of μ and σ^2 are computable in time $O(1)$.*

4 Poisson law for rare occurrences

Reinert and Schbath [11] showed that the number of occurrences of non overlapping finite sets of words is asymptotically Poisson, the total deviation error being $O(1/n)$ for texts of size n. On the other hand, Régnier and Szpankowski [10] give a full asymptotic expansion of the probability of finding r occurrences of a motif when r is $O(1)$, when the motif is a finite set of words of same lengths. Up to terms of order $O(1/n)$, the distribution of rare occurrences is Poisson. We adapt here a theorem of Régnier and Szpankowski [10] for the case of non self-overlapping and non nested languages. We re-write equation 3 as a series in the variable u:

$$F(z,u) = \frac{1}{1-z+L(z)} + \cdots + \frac{L(z)^k}{(1-z+L(z))^{k+1}} u^k + \dots. \tag{10}$$

Following a proof of Flajolet and *al.* [3], we show that the equation $1-z+L(z)=0$ has one real root ρ inside the open disk $|z|<2$, and that this root is unique inside this disk and greater than 1.

We consider a non nested and non self-overlapping language L. Let B and T respectively be the sets of first and last letters of the words of L. The language L is non self-overlapping and therefore we have $B \cap T = \emptyset$. Let $\pi_B = \sum_{\ell \in B} \pi_\ell$ be the sum of probability of the letters of B and π_T be the sum of probabilities of the letters of T. We have $L \subset B\Sigma^\star T \subseteq \Phi = B\Sigma^\star(\Sigma - B)$; the inclusion follows from the fact that if b is a letter of B and t a letter of T, the words bbt and btt of $B\Sigma T$ overlap. We have

$$\pi_B \pi_T \leq \max\left(\pi_B(1-\pi_B), \pi_T(1-\pi_T)\right) \leq \max_{0 \leq p \leq 1/2} p(1-p) = 1/4.$$

Therefore, for z real positive, the probability series $L(z)$ is strictly dominated by the probability series $\Phi(z)$ of the language Φ, which gives

$$L(z) < \Phi(z) \leq \pi_B z \frac{1}{1-z}(1-\pi_B)z \leq \frac{z^2}{4(1-z)}.$$

For $|z|=2$, we have $|L(z)|<1$ and $|1-z| \geq 1$. Therefore, Rouché theorem applies (see Jameson [5] or Bak and Newman [2]) and the functions $f = 1-z$ and $f+g = \phi(z) = 1-z+L(z)$ have the same number of zeroes inside the disk of radius 2, namely one. Let ρ be the zero of $1-z+L(z)$ in this disk. It is proved in [6] by use of the Perron-Frobenius theorem that the root $\rho(u)$ of smallest modulus of the equation $1-z+(1-u)L(z)=0$ is real and continuous in u. This implies that $\rho = \rho(0)$ is real. A probabilistic argument implies that ρ cannot be less than 1, and, at $z=1$, the function $1-z+L(z)$ is strictly positive. Therefore, we have $\rho > 1$. On the other hand, we consider a prefix-free language L. Let τ the (possibly infinite) radius of convergence of $L(z)$. The function $\phi(z)$ is infinite for $z \geq \tau$ and equal to zero for $z = \rho$. This implies that $1 < \rho < \tau$. With this at hand, the following theorem is a reformulation of Theorem 2.2 (ii) of Régnier and Szpankowski [10].

Theorem 3. *Let L be a non self-overlapping and non nested language with probability generating function $L(z)$ and $p_{n,k}$ be the probability that exactly k matches with L occur in a random text of size n.*
Then

1. *The generating function of words with k matches with L is*

$$L^{(k)}(z) = \frac{L(z)^k}{(1-z+L(z))^{k+1}}.$$

2. *There exists a real root ρ of smallest modulus and of multiplicity one of the equation $1-z+L(z)=0$; moreover, we have $\rho > 1$ and there exists $\kappa > \rho$ such that, for $k = O(1)$,*

$$p_{n,k} = \sum_{j=1}^{k+1} (-1)^j a_j \binom{n}{j-1} \rho^{-(n+j)} + O(\kappa^{-n}),$$

where

$$a_{k+1} = \frac{L(\rho)^k}{\rho^{k+1}(L'(\rho)-1)^{k+1}},$$

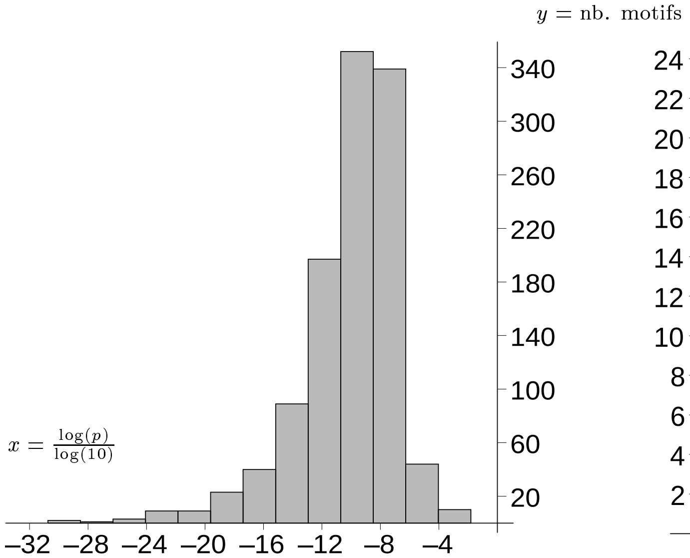

Figure 1: Histogram of probability of occurrence p of a motif at a given position for the 1118 motifs exactly computed. (ProDom distribution for amino-acids)

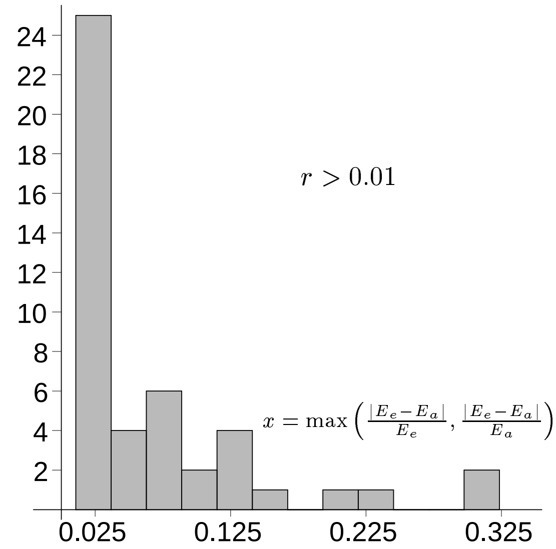

Figure 2: Rate of error r for the 1118 motifs exactly computed. E_e is the exact expectation, E_a is the approximate expectation on a sequence of length 6,756,720 positions with the ProDom amino-acids distribution

---

and the remaining coefficients verify

$$a_j = \frac{1}{(k+1-j)!} \lim_{z \to \rho} \frac{d^{k+1-j}}{dz^{k+1-j}} \left(L(z)^k \left(\frac{z-\rho}{1-z+L(z)} \right)^{k+1} \right).$$

The following corollary is an immediate consequence of this theorem.

Corollary 1. *Asymptotically, rare occurrences verify a Poisson law,*

$$p_{n,k} = \frac{\rho^{-n-1}}{1-L'(\rho)} \frac{(\alpha n)^k}{k!} \left(1 + O\left(\frac{1}{n}\right)\right), \qquad \alpha = \left(\frac{\rho^{-1} L(\rho)}{1-L'(\rho)}\right).$$

As a concluding remark of this section, the first error term in the corollary is k/n times the dominant term of the theorem. Practically, most of Prosite motifs have expectation under 10 for proteoms larger than one million amino-acids, and the Poisson law is a good approximation for these motifs. The expectation of the Poisson law is computable by Lemma 3.

5 Validation of the fast approximate method for protein motifs

For the 20 letters protein alphabet, the probability that an occurrence of a motif begins at a given position is small. This further implies that the probability of two overlapping matches of a motif, of order two with respect to this probability of occurrence, is much smaller and suggests that the non-overlapping approximation is good. On the other side, observations of Prosite motifs shows that the probability that a motif is nested is also very small. These hypotheses are confirmed by computations done with the exact method and verified as follows. Nicodème *et al.* [6] computed the expectation of 1118 Prosite motifs on the 6,756,720 amino-acids long non-redundant database of consensus of the multiple alignments of ProDom [16]. Figure 1 displays the probability of occurrences of these motifs at a given position of a random database of same length and same amino-acids distribution, where the probability of occurrence is computed as the ratio of the expectation by the length of the random database. This figure shows that

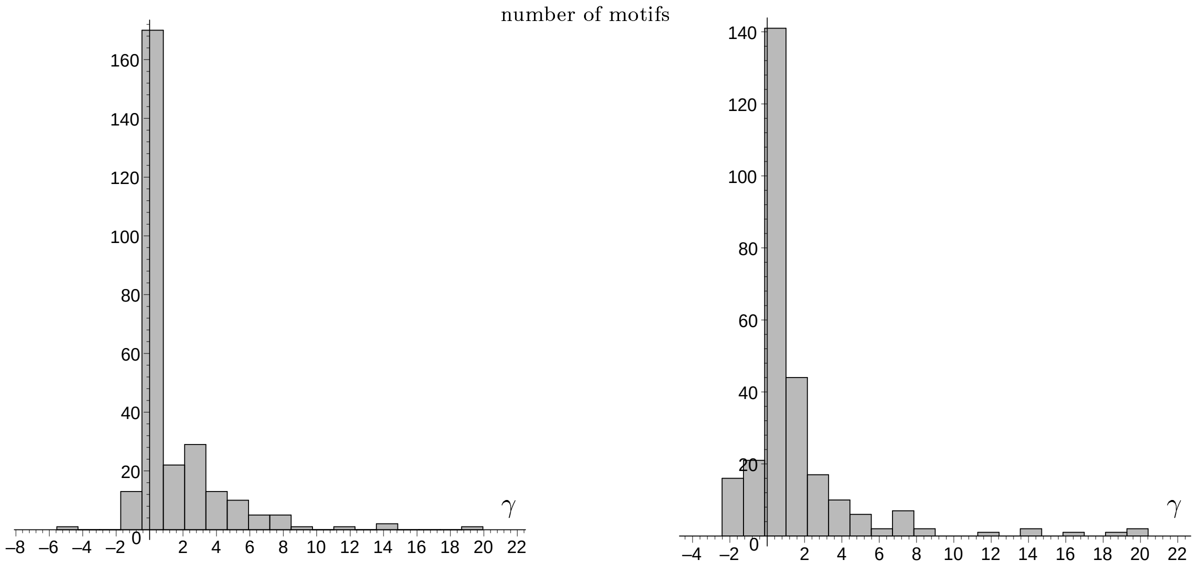

Figure 3: Histograms of the Gaussian quantiles γ of the PROSITE motifs for *S. cerevisiae* (left) and *C. elegans* (left) with expectation above 0.05 (273 motifs)

less than 1% of the motifs have probability above 10^{-4}. Figure 2 displays the rate of error

$$r = \max\left(\frac{|E_e - E_a|}{E_e}, \frac{|E_e - E_a|}{E_a}\right),$$

where E_e and E_a are respectively the exact expectation and the approximate expectation for ProDom. The display is limited to the 46 motifs (out of 1118) with $r > 0.01$. Out of these 46 motifs, 11 have a rate of error above 10% and the biggest error rate is 0.3. These results (99% of the motifs within a 10% error range and 96% within a 1% error range) are good in respect with the precision needed by biological applications. However, when the expectation of a motif is large, a 10% error would lead to misleading conclusions and the expectation must be computed exactly in this case.

We proceeded also to direct verification of the properties of motifs as follows. For each motif M that has been evaluated with the exact method, we examined the automaton recognizing $\Sigma^* M$. If, in this automaton, the transitions from final states, for any letter l of the alphabet, goes to the state accessed from the initial state with the same letter, the motif is not self-overlapping and not nested. This algorithmic check counts 406 non self-overlapping and non-nested motifs, about one third of the 1118 considered motifs. We also constructed an automaton for the finite language of words generated by the motif. From this automaton, we compute the generating function of the languages generated by the motifs and we compare it with the generating function computed by direct translation. This comparison disagrees for 7 PROSITE motifs that have ambiguous descriptions.

6 Genome comparison

This section is intended as an example of what can be done with the fast approximate method. A more detailed biological article is part of future work.

We compare here the 7,120,115 positions long proteom of *C. elegans* to the 4,049,041 positions long proteom of *S. cerevisiae*.

For each motif, and for each proteom, we computed with the approximate method the expectation μ of the number of matches. The letter frequencies that we use in the mathematical and the computational model are the empirical frequencies in each proteom, the quantities μ are determined by the `pattern2exp` program which implements the approximate method. When considering *C. elegans*, 38 motifs have expectation over 10. The expectation of 34 of these motifs has been exactly recomputed for the two organisms. The computation of the last four motifs is too long and these motifs have been left aside.

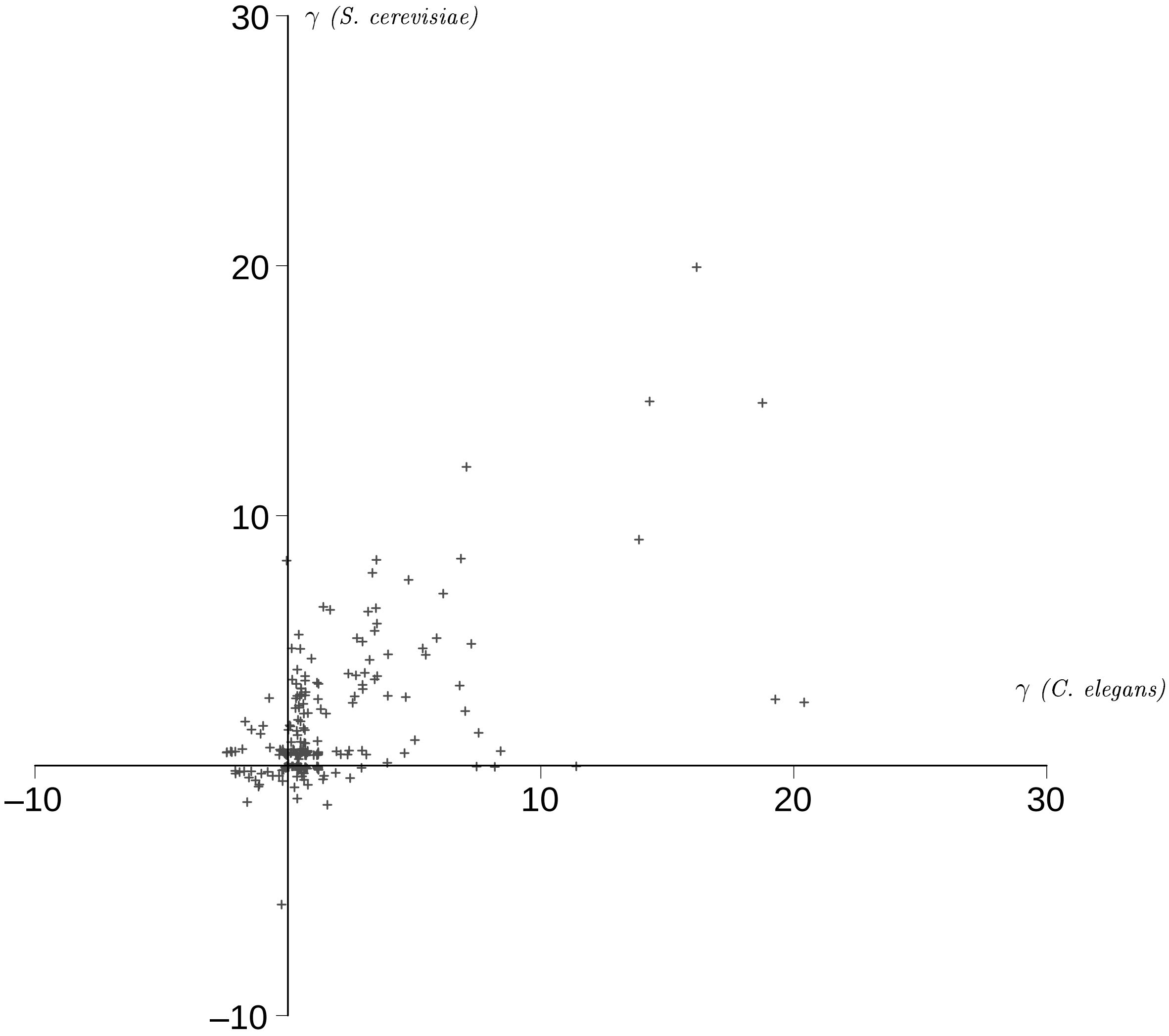

Figure 4: Gaussian quantiles of the PROSITE motifs for *S. cerevisiae* (x) and *C. elegans* (y) with expectation above 0.05 (273 motifs)

Each value for μ is then compared to the corresponding number of observed matches (also called observables), denoted by ω, that is obtained by a straight scan of the two proteoms[1]. To avoid discrepancies resulting of the different lengths of the two proteoms, the μ and ω values have been normalized to 1 million positions long proteoms and comparisons are made on these normalized values.

PROSITE motifs PS00001 to PS00009 are highly degenerate. These motifs and the few constrained motifs that have matches occurring only at the beginning or at the end of a text have been left out of the comparison.

A way to quantify the discrepancy between the expectation μ and the observation ω of the number of occurrences is by mean of the tail probability p of the asymptotic Poisson distribution and of the corresponding quantile of the normal distribution, for normalization. Let $x(p)$ be the solution of the

[1] The observed quantities were determined by the PROSITE tools contained in the IRSEC motif toolbox `http://www.isrec.isb-sib.ch/ftp-server/`. The computation of the observed values takes about 25 minutes for *C. elegans* and 15 minutes for *S. cerevisiae*.

```
PS00022 GC   20.41 GS    2.54 OC    430 EC       3.666 OS      4 ES        .321
PS00022
PS00022 C-x-C-x(5)-G-x(2)-C
PS00022 EGF_1;PATTERN.   EGF-like domain signature 1.

PS01186 GC   19.27 GS    2.66 OC    384 EC       3.260 OS      3 ES        .270
PS01186
PS01186 C-x-C-x(2)-[GP]-[FYW]-x(4,8)-C
PS01186 EGF_2;PATTERN.   EGF-like domain signature 2.

PS00010 GC   11.40 GS    -.03 OC    113 EC        .576 OS      0 ES        .061
PS00010
PS00010 C-x-[DN]-x(4)-[FY]-x-C-x-C
PS00010 ASX_HYDROXYL;PATTERN.   Aspartic acid and asparagine hydroxylation site.

PS00109 GC    8.42 GS     .58 OC     81 EC       1.306 OS      2 ES        .570
PS00109
PS00109 [LIVMFYC]-x-[HY]-x-D-[LIVMFY]-[RSTAC]-x(2)-N-[LIVMFYC](3)
PS00109 PROTEIN_KINASE_TYR;PATTERN.   Tyrosine protein kinases specific active-site signature.

PS00232 GC    8.18 GS    -.04 OC     53 EC        .168 OS      0 ES        .103
PS00232
PS00232 [LIV]-x-[LIV]-x-D-x-N-D-[NH]-x-P
PS00232 CADHERIN;PATTERN.   Cadherins extracellular repeated domain signature.

PS00018 GC    7.55 GS    1.32 OC    184 EC      24.606 OS     30 ES      20.373
PS00018
PS00018 D-x-[DNS]-{ILVFYW}-[DENSTG]-[DNQGHRK]-{GP}-[LIVMC]-[DENQSTAGC]-x(2)-[DE]-[LIVMFYW]
PS00018 EF_HAND;PATTERN.   EF-hand calcium-binding domain.

PS00280 GC    7.47 GS    -.04 OC     62 EC        .721 OS      0 ES        .088
PS00280
PS00280 F-x(3)-G-C-x(6)-[FY]-x(5)-C
PS00280 BPTI_KUNITZ;PATTERN.   Pancreatic trypsin inhibitor (Kunitz) family signature.
```

	A	R	N	D	C	Q	E	G	H	I
C. El.	.0622	.0513	.0496	.0523	.0207	.0408	.0641	.0531	.0232	.0624
S. Ce.	.0557	.0444	.0612	.0580	.0129	.0392	.0648	.0499	.0217	.0656
	L	K	M	F	P	S	T	W	Y	V
C. El.	.0870	.0644	.0262	.0497	.0484	.0804	.0586	.0111	.0322	.0623
S. Ce.	.0956	.0729	.0209	.0448	.0438	.0898	.0589	.0104	.0336	.0561

Figure 5: (Above). Motifs that are functional in *C. elegans* and not functional in *S. cerevisiae*. GC, OC, EC and GS, OS, ES respectively are the Gaussian quantiles γ, the observations ω and the expectations μ of the number of occurrences for *C. elegans* and *S. cerevisiae*; the Gaussian quantiles have been computed for equivalent (in the sense of the amino-acid distribution) 10^6 amino-acids long databases. (Below). Probability of occurrence of the amino-acids in the two organisms.

---

equation

$$\int_x^\infty \frac{1}{\sqrt{2\pi}} e^{-t^2/2} dt = p.$$

The signed quantile γ calibrating the observed number of occurrences is computed along the following rules

$$\text{Case} \quad \omega > \mu : \qquad p = \sum_{k \geq \omega} e^{-\mu} \frac{\mu^k}{k!} = 1 - \sum_{0 \leq k < \omega} e^{-\mu} \frac{\mu^k}{k!.}, \qquad \gamma = +x(p).$$

$$\text{Case} \quad \omega < \mu : \qquad p = \sum_{0 \leq k \leq \omega} e^{-\mu} \frac{\mu^k}{k!}, \qquad \gamma = -x(p).$$

We apply this discrepancy measure in the following section.

Results

Figure 3 displays histograms of the Gaussian quantiles γ for the multi-cellular organism *C. elegans* and the mono-cellular organism *S. cerevisiae*. Recall that the tail probability for quantiles $\gamma = 5$ and $\gamma = 6$ respectively are less than 3.10^{-7} and 10^{-9}. Such quantiles correspond to exceptional events. Figure 4 compares the quantiles computed for 273 motifs for the two organisms.

We focus on the motifs that have high quantile values γ for *C. elegans* but small ones for *S. cerevisiae*.

Both PS00022 and PS01186 are Epidermal Growth Factors that are statistically insignificant in *S. cerevisiae* while being functional during the development in *C. elegans*.

PS00010 occurs in several proteins of *C. elegans*, such as GPL1_CAEEL that is implicated in cell-cell interactions, and does not occur in *S. cerevisiae* where there is no hydroxylation.

PS00109 is a Tyrosine kinase protein. The biological mechanism of addition of phosphate to Tyrosine is typical of multi-cellular organisms.

The cadherins extracellular PS00232 is involved in the assembly of cells in multi-cellular organisms and does not occur in *S. cerevisiae*.

The EF-hand calcium-binding domain PS00018 is strongly over-represented in *C. elegans* and slightly in *S. cerevisiae*; this suggests that the calmodulin-based regulatory mechanism is much more elaborated in *C. elegans*.

PS00280 is a pancreatic enzyme that does not appear in the mono-cellular *S. cerevisiae*.

7 Implementation and performances

The fast approximate method is implemented as a AWK program (`pattern2exp`). This program [2] together with MAPLE software and worksheets for the exact method is available at address "`http://www.dkfz-heidelberg.de/tbi/people/nicodeme/regexp`". The expectations of the number of occurrences of motifs according to an amino-acid distribution and to a database length are computed. The present version only handles protein motifs.

The performance is independent of the length of the database and is only related to the number of input motifs. The 1260 non-constrained PROSITE motifs are processed in about 30 seconds. It takes 30 minutes to make the exact computation for the 34 motifs for whom the approximation is too loose for *C. elegans*. As a comparison, the exact method takes about 30 computation days and computes 88% of the motifs, the 142 left motifs exceeding a time limit of 4 hours.

8 Conclusion and future work

We presented in this article a fast approximate method that calibrates the motifs of PROSITE for any amino-acids distribution. We apply the method to compare *C. elegans* and *S. cerevisiae*, and we find motifs that are functional in *C. elegans* but not in *S. cerevisiae*. The probabilistic calibration of occurrences of motifs performed here for these two organisms may be done for other organisms. Exceptional quantiles in the calibration clearly indicate which motifs are functional in some organisms and are not in other ones. They provide strong suggestions for further biological investigations.

We have not found yet a theoretical method to bound the error done when the motif considered is self-overlapping. This occurs frequently for DNA motifs, and therefore the approximate method considered here is not presently adapted for this latter case. Future research should tackle this problem. Future work also includes an integrated genome comparison package for motifs and detailed biological analysis of genome comparisons based on probabilistic calibration of occurrences of PROSITE motifs.

Acknowledgements

We thank an anonymous referee for helpful comments.

[2] See also the `regexpcount` program of the `algolib` software library at address `http://algo.inria.fr/libraries/software.html`.

References

[1] Atteson, K. Calculating the exact probability of language-like patterns in biomolecular sequences. In *Sixth International Conference on Intelligent Systems for Molecular Biology* (1998), AAAI Press, pp. 17–24.

[2] Bak, J., and Newman, D. J. *Complex Analysis.* Springer, 1997.

[3] Flajolet, P., Gourdon, X., and Martínez, C. Patterns in Random Binary Search Trees. *Random Structures and Algorithms 11*, 3 (1997), 223–244.

[4] Flajolet, P., and Sedgewick, R. The average case analysis of algorithms: Counting and generating functions. Research Report 1888, Institut National de Recherche en Informatique et en Automatique, Apr. 1993.

[5] Jameson, G. J. *A First Course on Complex Functions.* Chapman and Hall, 1970.

[6] Nicodème, P., Salvy, B., and Flajolet, P. Motif statistics. Research Report 3606, Institut National de Recherche en Informatique et en Automatique, Jan. 1999. Accepted by ESA99, see "http://pauillac.inria.fr/algo/papers/".

[7] Pevzner, P. A., Borodovski, M. Y., and Mironov, A. A. Linguistic of nucleotide sequences: The significance of deviation from mean statistical characteristics and prediction of the frequencies of occurrence of words. *J. Biomol. Struct. Dyn 6* (1989), 1013–1026.

[8] Prum, B., Rodolphe, F., and de Turckheim, E. Finding words with unexpected frequencies in deoxyribonucleic acid sequences. *J. R. statist. Soc. B 57*, 1 (1995), 205–220.

[9] Régnier, M. A unified approach to words statistics. In *Second Annual International Conference on Computational Molecular Biology* (1998), ACM Press, New-York, pp. 207–213.

[10] Régnier, M., and Szpankowski, W. On Pattern Frequency Occurrences in a Markovian Sequence. *Algorithmica 22*, 4 (1998), 631–649.

[11] Reinert, G., and Schbath, S. Compound Poisson Approximations for Occurrences of Multiple Words in Markov Chains. *J. Comp. Biol. 5*, 2 (1998), 223–253.

[12] Salomaa, A., and Soittola, M. *Automata-Theoretic Aspects of Formal Power Series.* Springer, 1978. Texts and monographs in computer science.

[13] Schbath, S., Prum, B., and de Turckheim, É. Exceptional motifs in different Markov chain models for a statistical analysis of DNA sequences. *J. Comp. Biol. 2*, 3 (1995), 417–437.

[14] Sedgewick, R., and Flajolet, P. *An Introduction to the Analysis of Algorithms.* Addison-Wesley Publishing Company, 1996.

[15] Sewell, R. F., and Durbin, R. Method for calculation of probability of matching a bounded regular expression in a random data string. *J. Comp. Biol. 2*, 1 (1995), 25–31.

[16] Sonnhamer, E. L. L., and Kahn, D. The modular arrangement of proteins as inferred from analysis of homology. *Protein Science 3* (1994), 482–492. "http://protein.toulouse.inra.fr/prodom.html".

[17] Titchmarsh, E. *The theory of functions.* Oxford Science Publications, 1939. Second edition.