Monotony of Surprise And Large-Scale Quest for Unusual Words

(Extended Abstract)

Alberto Apostolico* *Univ. di Padova & Purdue Univ.*

Mary Ellen Bock† *Purdue University*

Stefano Lonardi‡ *Univ. of California, Riverside*

ABSTRACT

The problem of characterizing and detecting recurrent sequence patterns such as substrings or motifs and related associations or rules is variously pursued in order to compress data, unveil structure, infer succinct descriptions, extract and classify features, etc. In Molecular Biology, exceptionally frequent or rare words in bio-sequences have been implicated in various facets of biological function and structure. The discovery, particularly on a massive scale, of such patterns poses interesting methodological and algorithmic problems, and often exposes scenarios in which tables and synopses grow faster and bigger than the raw sequences they are meant to encapsulate. In previous study, the ability to succinctly compute, store, and display unusual substrings has been linked to a subtle interplay between the combinatorics of the subwords of a word and local monotonicities of some scores used to measure the departure from expectation. In this paper, we carry out an extensive analysis of such monotonicities for a broader variety of scores. This supports the construction of data structures and algorithms capable of performing global detection of unusual substrings in time and space linear in the subject sequences, under various probabilistic models.

*Corresponding author. Dipartimento di Elettronica e Informatica, Università di Padova, Padova, Italy, and Department of Computer Sciences, Purdue University, Computer Sciences Building, West Lafayette, IN 47907, USA. Work supported in part by NSF Grant CCR-9700276, by Purdue Research Foundation Grant 690-1398-3145, by the Italian Ministry of University and Research under the National Project "Bioinformatics and Genomics Research", and by the Research Program of the University of Padova. `axa@dei.unipd.it`

†Department of Statistics, Purdue University, West Lafayette, IN 47907, USA. `mbock@stat.purdue.edu`

‡Department of Computer Science and Engineering, University of California, Riverside, CA 92521, USA. Work supported by Purdue Research Foundation Grant 690-1398-3145, by the Italian Ministry of University and Research under the National Project "Bioinformatics and Genomics Research", and by the Research Program of the University of Padova. `stelo@cs.ucr.edu`

1. INTRODUCTION AND SUMMARY

Words that occur unexpectedly often or rarely in genetic sequences have been variously linked to biological meanings and functions. The underlying probabilistic and statistical models have been studied extensively and led to the production of a rich mass of results (see, e.g., [16, 18]). With increasing availability of whole genomes, exhaustive statistical tables and global detectors of unusual words on a scale of millions, even billions of bases become conceivable. It is natural to ask how large such tables may grow with increasing length of the input sequence, and how fast they can be computed. These problems need to be regarded not only from the conventional perspective of asymptotic space and time complexities, but also in terms of the volumes of data produced and ultimately, of practical accessibility and usefulness. Tables that are too large at the outset saturate the perceptual bandwidth of the user, and might suggest approaches that sacrifice some modeling accuracy in exchange for an increased throughput. The focus of the present paper is thus on the combinatorial structure of such tables and on the algorithmic aspects of their implementation. To make our point more clear, we discuss here the problem of building exhaustive statistical tables for *all* subwords of very long sequences. But it should become apparent that reflections of our arguments are met just as well in most practical cases.

The number of distinct substrings in a string is at worst quadratic in the length of that string. Thus, the statistical table of all words for a sequence of a modest 1,000 bases may reach in principle into the hundreds of thousands of entries. Such a synopsis would be asymptotically bigger than the phenomenon it tries to encapsulate or describe. This is even worse than what the (now extinct) cartographers did in the old Empire narrated by Borges' fictitious J. A. Suàrez Miranda [7]: there, "Cartography attained such perfection that ... the College of Cartographers evolved a Map of the Empire that was of the same scale as the Empire and that coincided with it point for point[1]".

[1]Attributed to "Viajes de Varones Prudentes (Libro Cuarto, Cap. XLV, Lerida, 1658)", the piece "On the Exactitude of Science" was written in actuality by Jorge Luis Borges and

The situation does not improve if we restrict ourselves to computing and displaying the most *unusual* words in a given sequence. This presupposes that we compare the frequency of occurrence of every word in that sequence with its expectation: a word that departs from expectation beyond some preset *threshold* will be labeled as *unusual* or *surprising*. Departure from expectation is assessed by a distance measures often called a *score* function. The typical format for a z-score is that of a difference between observed and expected counts, usually normalized to some suitable moment. For most *a priori* models of a source, it is not difficult to come up with extremal examples of *observed* sequences in which the number of, say, over-represented substrings grows itself with the square of the sequence length: in such an empire, a map pinpointing salient points of interest would be bigger than the empire itself. Extreme as these examples might be, they do suggest that large statistical tables may not only be computationally imposing but also impractical to visualize and use, thereby defying the very purpose of their construction. In fact, similar risks are faced in the broad area of pattern and association discovery [1].

In this paper, we study probabilistic models and scores for which the population of potentially unusual words in a sequence can be described by tables of size at worst linear in the length of that sequence. This not only leads to more palatable representations for those tables, but also supports (non-trivial) linear time and space algorithms for their constructions. Note that these results do not mean that now the number of unusual words must be linear in the input, but just that their representation and detection can be made such. The ability to succinctly compute, store, and display our tables rests on a subtle interplay between the combinatorics of the subwords of a sequence and the monotonicity of some popular scores within small, easily describable classes of related words. Specifically, it is seen that it suffices to consider as candidate surprising words only the members of an *a priori* well identified set of "representative" words, where the cardinality of that set is linear in the text length. By the representatives being identifiable *a priori* we mean that they can be known before any score is computed. By neglecting the words other than the representatives we are not ruling out that those words might be surprising. Rather, we maintain that any such word: (i) is embedded in one of the representatives, and (ii) does not have a bigger score or degree of surprise than its representative (hence, it would add no information to compute and give its score explicitly).

As mentioned, a crucial ingredient for our construction is that the score be monotonic in each class. In this paper, we perform an extensive analysis of models and scores that fulfill such a monotonicity and are thus susceptible to this treatment. The main results comes in form of a series of conditions and properties, which we describe here without proofs within a framework primarily aimed at clarifying their significance and scope.

The paper is organized as follows. Section 2 describes some preliminary notation and properties. The monotonicity results are presented in tabular form in Section 3. Because the collection of proofs and supporting combinatorial lemmas are rather lengthy and technically involved, they could only be included in the full version of the paper. However, they shall be made available upon request. Finally, we briefly discuss the algorithmic implications and constructs in Section 4. We also highlight future work, and the extension of succinct descriptors of the kind considered here to more general models and outside of the monotonicity realm. These results are being incorporated into an existing suite of programs [12, 5]. As an example demonstration, Figure 1 shows application of the tool to the identification of the core modules within the regulatory regions of the yeast. Finding such modules is the first step towards a full-fledged promoter analytic system, which would help biologists to understand and investigate the gene expression in relation to development, tissue specificity and/or environment. Each one of the two families contains a set of co-regulated genes, that is, genes that have similar expression under the same external conditions. The hypothesis is that in each family the upstream region will contain some common motifs, and also that such signals might be over-represented across the family. In this, like in the countless other applications of probabilistic and statistical sequence analysis, access to the widest repertoire of models and scores is the crucial asset in the formulation, test and fine tuning of hypotheses.

2. PRELIMINARIES

We use standard concepts and notation about strings, for which we refer to [2, 3, 4]. For a substring y of a text x, we denote by $f(y)$ the number of occurrences of y in x. Clearly, for any *extension* uyv of y, $f(uyv) \leq f(y)$. For a set of strings or *multisequence* $\{x^{(1)}, x^{(2)}, \ldots x^{(k)}\}$, the *colors* of y are the members of the subset of the multisequence such that each contains at least one occurrence of y. The number of colors of y is denoted by $c(y)$. We also have $c(uyv) \leq c(y)$.

Suppose now that string $x = x_{[1]}x_{[2]}\ldots x_{[n]}$ is a realization of a stationary random process and $y_{[1]}y_{[2]}\ldots y_{[m]} = y$ is an arbitrary but fixed pattern over Σ with $m < n$. We define Z_i, for all $i \in [1 \ldots n-m+1]$, to be 1 if y occurs in x starting at position i, 0 otherwise, so that

$$Z_y = \sum_{i=1}^{n-m+1} Z_i$$

is the random variable for $f(y)$.

Expressions for the expectation and variance for the number of occurrences in the Bernoulli model, have been given by several authors (see, e.g., [9, 10, 14, 15, 17, 10, 9, 15]). Here we adopt derivations in [2, 3]. With p_a the probability of symbol $a \in \Sigma$ and $\hat{p} = \prod_{i=1}^{m} p_{y_{[i]}}$, we have

$$E(Z_y) = (n-m+1)\hat{p}$$

$$Var(Z_y) = \begin{cases} (1-\hat{p})E(Z_y) - \hat{p}^2(2n-3m+2)(m-1) \\ \quad +2\hat{p}B(y) & \text{if } m \leq (n+1)/2 \\ (1-\hat{p})E(Z_y) - \hat{p}^2(n-m+1)(n-m) \\ \quad +2\hat{p}B(y) & \text{otherwise} \end{cases}$$

Adolfo Bioy Casares. English translation quoted from [7]: "... succeeding generations came to judge a map of such magnitude cumbersome, and, not without irreverence, they abandoned it to the rigours of Sun and Rain ... in the whole Nation, no other relic is left of the Discipline of Geography."

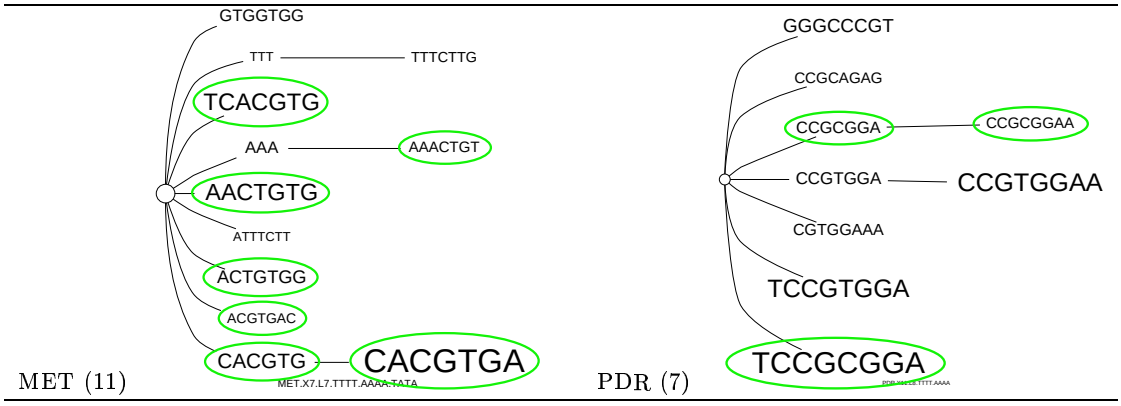

Figure 1: Over-represented words in a set of co-regulated genes. A word's increasing departure from its expected frequency is rendered by proportionally increased font size. Superposition of the words circled by hand yields the previously known motifs: `TCACGTG` and `AAAACTGTGG` in the MET family of 11 sequences, and `TCCGCGGA` in the PDR family of 7.

where

$$B(y) = \sum_{d \in \mathcal{P}(y)} (n - m + 1 - d) \prod_{j=m-d+1}^{m} p_{y_{[j]}}$$

is the *auto-correlation factor* of y, that depends on the set $\mathcal{P}(y)$ of the lengths of the periods[2] of y. In cases of practical interest we expect $m \leq (n+1)/2$, so that we make this assumption from now on.

In the case of Markov chains, it is more convenient to evaluate the estimator of the expectation instead of the true expectation to avoid computing large transition matrices. In fact, we can estimate the expected number of occurrences in the M-order Markov model with the following maximum likelihood estimator [16]

$$\begin{aligned} \hat{E}(Z_y) &= \frac{\prod_{i=1}^{m-M} f(y_{[i,i+M]})}{\prod_{i=2}^{m-M} f(y_{[i,i+M-1]})} \\ &= f(y_{[1,M+1]}) \prod_{j=2}^{m-M} \frac{f(y_{[j,j+M]})}{f(y_{[j,j+M-1]})} \end{aligned} \qquad (1)$$

The expression for the variance $Var(Z_y)$ for Markov chains is very involved. Complete derivations have been given by Lundstrom [13], Kleffe and Borodovsky [10], and Regnier and Szpankowski [15]. However, as soon as the true model is unknown and the transition probabilities have to be estimated from the observed sequence x the results for the exact distribution are no longer useful (see, e.g. [16]). In fact, once we replace the expectation with an *estimator* of the expected count, the variance of the difference between observed count and the estimator does not correspond anymore to the variance of the random variable describing the count.

The asymptotic variance of $E(Z_y) - \hat{E}(Z_y)$ has been given first by Lundstrom [13], and is clearly different from the asymptotic variance of $E(Z_y)$ (see [18] for a detailed exposition). Easier ways to compute the asymptotic variance were also found subsequently.

For a finite family $\{x^{(1)}, x^{(2)}, \ldots x^{(k)}\}$ of realizations of our process, and a pattern y, we analogously define W_j, for all $j \in [1 \ldots k]$, to be 1 if y occurs at least once in $x^{(j)}$, 0 otherwise. Let

$$W_y = \sum_{j=1}^{k} W_j$$

so that W_y is a random variable for the total number $c(y)$ of sequences which contain at least one occurrence of y.

In the case of a multisequence we can assume in actuality either a single model for the entire family or a distinct model for each sequence. In any case, the expectation of the random variable W for the number of colors can be computed by assuming a Poisson distribution as follows

$$E(W_y) = k - \sum_{j=1}^{k} e^{-E(Z_y^j)} \qquad (2)$$

where $E(Z_y^j)$ is the expected number of occurrences of the word y in the j-th sequence [17].

Ideally, a score function should be independent of the structure and size of the word. That would allow one to make meaningful comparisons among substrings of various compositions and lengths based on the value of the score.

There is some general consensus that z-scores may be preferred over the others [11]. For any word w, a standardized frequency called z-score, can be defined by

$$z(y) = \frac{f(y) - E(Z_y)}{\sqrt{Var(Z_y)}}$$

If $E(Z_y)$ and $Var(Z_y)$ are known, then under rather general conditions, the statistics $z(y)$ is asymptotically normally distributed with zero mean and unit variance as n tends to infinity. In practice $E(Z_y)$ and $Var(Z_y)$ are seldom known, but are estimated from the sequence under study.

[2] String z has a *period* w if z is a non-empty prefix of w^k for some integer $k \geq 1$.

For a given type of count and model, we consider now the problem of computing exhaustive tables reporting scores for all substrings of a sequence, or perhaps at least for the most surprising among them. The problem comes in different flavors based on the probabilistic model. However, a table for all words of any size would require quadratic space in the size of the input, not to mention that such a table would take at least quadratic time to be filled.

As seen towards the end of the paper such a limitation can be achieved by partitioning the set of all words into equivalence classes with the property, that it suffices to account for only one or two candidate surprising words in each class, while the number of classes is linear in the textstring size. More formally, given a score function z, a set of words C, and a real positive *threshold* T, we say that a word $w \in C$ is *T-over-represented* in C (resp., *T-under-represented*) if $z(w) > T$ (resp., $z(w) < -T$) and for all words $y \in C$ we have $z(w) \geq z(y)$ (resp., $z(w) \leq z(y)$). We say that a word w is *T-surprising* if $z(w) > T$ or $z(w) < -T$. We also call $\max(C)$ and $\min(C)$ respectively the longest and the shortest word in C, when $\max(C)$ and $\min(C)$ are unique.

Let now x be a textstring and $\{C_1, C_2, \ldots, C_l\}$ a partition of all its substrings, where $\max(C_i)$ and $\min(C_i)$ are uniquely determined for all $1 \leq i \leq l$. For a given score z and a real positive constant T, we call $\mathcal{O}_z^T$ the set of T-over-represented words of C_i, $1 \leq i \leq l$, with respect to that score function. Similarly, we call $\mathcal{U}_z^T$ the set of T-under-represented words of C_i, and $\mathcal{S}_z^T$ the set of all T-surprising words, $1 \leq i \leq l$.

For two strings v and $u = svz$, a *(u, v)-path* is a sequence of words $\{w_0 = u, w_1, w_2, \ldots, w_j = v\}$, $l \geq 0$, such that w_i is a unit-symbol extension of w_{i-1} $(1 \leq i \leq j)$. In general a (u, v)-path is not unique. If all $w \in C$ belong to some $(min(C_i), max(C_i))$-path, we say that class C is *closed*.

A score function z is *(u, v)-increasing* (resp., *non-decreasing*) if given any two words w_1, w_2 belonging to a (u, v)-path, the condition $|w_1| < |w_2|$ implies $z(w_1) < z(w_2)$ (resp., $z(w_1) \leq z(w_2)$). The definitions of a *(u, v)-decreasing* and *(u, v)-non-increasing* z-scores are symmetric. We also say that a score z is *(u, v)-monotonic* when specifics are unneeded or understood. The following fact and its symmetric are immediate.

FACT 2.1. *If the z-score under the chosen model is $(min(C_i), max(C_i))$-increasing, and C_i is closed, $1 \leq i \leq l$, then*

$$\mathcal{O}_z^T \subseteq \bigcup_{i=1}^{l} \max(C_i) \quad \textit{and} \quad \mathcal{U}_z^T \subseteq \bigcup_{i=1}^{l} \min(C_i)$$

Some scores are defined in terms of the absolute value (or any even power) of a function of expectation and count. In those cases, we cannot distinguish anymore over-represented from under-represented words. This restriction is compensated by the fact that we can now relax the property asked of the score function, as explained next.

We recall that a real-valued function F is *concave* in a set S of real numbers if for all $x_1, x_2 \in S$ and all $\lambda \in (0, 1)$ we have $F((1 - \lambda)x_1 + \lambda x_2) \geq (1 - \lambda)F(x_1) + \lambda F(x_2)$. If F is concave, then the set of points below its graph is a concave set. Also, given two functions F and G such that F is concave and G is concave and monotonically decreasing, we have that $G(F(x))$ is concave.

Similarly, a function F is *convex* in a set S if for all $x_1, x_2 \in S$ and all $\lambda \in (0, 1)$ we have $F((1 - \lambda)x_1 + \lambda x_2) \leq (1 - \lambda)F(x_1) + \lambda F(x_2)$. If F is convex, then the set of points above its graph is a convex set. Also, given two functions F and G such that F is convex and G is convex and monotonically increasing, we have that $G(F(x))$ is convex.

FACT 2.2. *If the z-score under the chosen model is a convex function of a $(min(C_i), max(C_i))$-monotonic score z', that is*

$$z((1 - \lambda)z'(u) + \lambda z'(v)) \leq (1 - \lambda)z(z'(u)) + \lambda z(z'(v))$$

for all $u, v \in C_i$, and C_i is closed, $1 \leq i \leq l$, then

$$\mathcal{S}_z^T \subseteq \bigcup_{i=1}^{l} \{\max(C_i) \cup \min(C_i)\}$$

This fact has two useful corollaries.

COROLLARY 2.1. *If the z-score under the chosen model is the absolute value of a score z' which is $(min(C_i), max(C_i))$-monotonic, and C_i is closed, $1 \leq i \leq l$, then*

$$\mathcal{S}_z^T \subseteq \bigcup_{i=1}^{l} \{\max(C_i) \cup \min(C_i)\}$$

COROLLARY 2.2. *If the z-score under the chosen model is a convex and increasing function of a score z', which is in turn a convex function of a score z'' which is $(min(C_i), max(C_i))$-monotonic, and C_i is closed, $1 \leq i \leq l$, then*

$$\mathcal{S}_z^T \subseteq \bigcup_{i=1}^{l} \{\max(C_i) \cup \min(C_i)\}$$

An example to which the latter corollary could be applied is the choice $z = (z')^2$ and $z' = |z''|$.

Sometimes we are interested in finding words which *minimize* the value of a positive score instead of maximizing it. A fact symmetric to Fact 2.2 also holds.

FACT 2.3. *If the z-score under the chosen model is a concave function of a $(min(C_i), max(C_i))$-monotonic score z', that is*

$$z((1 - \lambda)z'(u) + \lambda z'(v)) \geq (1 - \lambda)z(z'(u)) + \lambda z(z'(v))$$

for all $u, v \in C_i$, and C_i is closed, $1 \leq i \leq l$, then the set of words for which the z-score is minimized is contained in

$$\bigcup_{i=1}^{l} \{\max(C_i) \cup \min(C_i)\}$$

In the next section we present monotonicities established for a number of scores for words w and wv that obey a condition of the form $f(w) = f(wv)$, i.e., have the same set of occurrences. In Section 4 we discuss in more detail some of the partitions induced by such a condition with a linear number of equivalence classes.

3. MONOTONICITY RESULTS

The tables of this section display a collection of monotonicity results established about the models and z-scores considered. The corresponding proofs are quite lengthy, some involve several auxiliary lemmas on the combinatorics of subwords. Thus they are deferred to the full paper. Here we limit ourselves to very few comments aimed at illustrating the properties.

Throughout, we assume w and an extension wv of w to be nonempty substrings of a text x such that $f(w) = f(wv)$. For convenience of notation, we set $\rho(w) \equiv E(w)/N(w)$, where $N(w)$ appears in the score as the expected value of some function of w. The interpretation of the tables is straightforward. For example, Property 1.1 states a simple fact on the monotonicity of $E(w)$ given the monotonicity of $\rho(w)$ and $N(w)$. Under some general conditions on $N(w)$ and $\rho(w)$ we can prove the monotonicity of any score functions of the form described above.

Property 1.2 is not straightforward. It says that these scores are monotonically decreasing when

$$f < E^* = E(w)\frac{\gamma N(w) + N(wv)}{N(w) + N(wv)}$$

and monotonically increasing when $f > E^*$. We can picture the dynamics of the score as follow. Initially, we can assume $E^* > f$, in which case the score is decreasing. As we extend the word, keeping the count f constant, E^* decreases (recall that E^* is always in the interval $[E(wv), E(w)]$). At some point, $E^* = f$, in which case the score stays constant. By extending the word even more, E^* becomes smaller than f, and the score starts to grow. Some consequences of Property 1.2 are captured by Properties 1.7 and 1.8. Property 1.2 also holds by exchanging the condition $\rho(wv) \leq \rho(w)$ with $f(w) > E(w) > E(wv)$.

As mentioned, certain types of scores require to be minimized rather than maximized. For example, the scores based on the probability that $\mathbf{P}(f(w) \leq T)$ or $\mathbf{P}(f(w) \geq T)$ for a given threshold T on the number of occurrences. One can prove then

FACT 3.1. *Given a threshold $T > 0$ on the number of occurrences, then*

$$\boldsymbol{P}(f(w) \leq T) \leq \boldsymbol{P}(f(wv) \leq T)$$

Let us consider the score $z_P(w,T) = \min\{\mathbf{P}(f(w) \leq T), \mathbf{P}(f(w) > T)\} = \min\{\mathbf{P}(f(w) \leq T), 1 - \mathbf{P}(f(w) \leq T)\}$ evaluated on the strings in a class C. By the above Fact one can compute the score only for the shortest and the longest string in C, as $\min\{\mathbf{P}(f(\min(C)) \leq T), \mathbf{P}(f(\max(C)) > T)\}$. The score $z_P(w,T)$ satisfies also the conditions of Fact 2.3. In fact, $z' = \mathbf{P}(f_x(w) \leq T)$ is $(min(C), max(C))$-monotonic by Fact 3.1 and the transformation $z = \min\{z', 1 - z'\}$ is a concave function in z'.

Turning now to Table 2, we recall that p_a is the probability of the symbol $a \in \Sigma$ in the Bernoulli model. We define $\hat{p} = \prod_{i=1}^{|w|} p_{w_{[i]}}$ and $\hat{q} = \prod_{i=1}^{|v|} p_{v_{[i]}}$. Note that $0 < p_{min}^{|w|} \leq \hat{p} \leq p_{max}^{|w|} < 1$, where $p_{min} = \min_{a \in \Sigma} p_a$ and $p_{max} = \max_{a \in \Sigma} p_a$.

The lengthiest arguments here concern properties that involve the complete variance, as they must be based in turn on proofs of combinatorial properties of the autocorrelation function $B(w)$ introduced earlier. These monotonicities are reported under 2.9-2.12. As an example of an intermediate result, it is proved that given a word of size m, if $p_{max} \leq 1/\sqrt[m]{4m}$, the variance is monotonically decreasing for any choice of n and p_b, where b is the symbol added to the string. A slightly better bound on p_{max} can be attained numerically by considering that $p_b \leq p_{max}$.

Table 3 reports monotonicities derived properties for a Markov process of order $M > 0$. Monotonicities based on color-count under two models are similarly reported in Table 4.

4. COMPUTING EQUIVALENCE CLASSES AND SCORES

We recall the properties of a partition $\{C_1, C_2, \ldots, C_l\}$ of the substrings which would enable us to restrict the computation of the scores to a constant number of candidates in each class C_i. Namely, we require, for all $1 \leq i \leq l$, $\max(C_i)$ and $\min(C_i)$ to be unique; C_i to be closed, i.e., all w in C_i belong to some $(min(C_i), max(C_i))$-path; all w in C_i have the same count. Of course, the partition of all substrings of x into singleton classes fulfills those properties. In practice, we want l to be as small as possible.

We say that two strings y and w are *left-equivalent* on x if the set of starting positions of y in x matches the set of starting positions of w in x. We denote this equivalence relation by $\equiv_l$. It follows from the definition that if $y \equiv_l w$, then either y is a prefix of w, or vice versa. Therefore, each class has unique shortest and longest member. Also by definition, if $y \equiv_l w$ then $f(y) = f(w)$ and $c(y) = c(w)$. We similarly say that y and w are *right-equivalent* on x if the set of ending positions of y in x matches the set of ending positions of w in x. We denote this by $\equiv_r$.

Finally, the equivalence relation $\equiv_x$ is defined in terms of the *implication* of a substring of x [6, 8]. Given a substring w of x, the implication $imp_x(w)$ of w in x is the longest string uwv such that every occurrence of w in x is preceded by u and followed by v. We write $y \equiv_x w$ iff $imp_x(y) = imp_x(w)$. It is not difficult to see that the equivalence relation $\equiv_x$ is the transitive closure of $\equiv_l \cup \equiv_r$. Well known results (see, e.g., [6, 8]) show that the size l of the partition is linear in $|x| = n$ for all three equivalence relations considered. In particular, the smallest size is attained by $\equiv_x$, for which the number of equivalence classes is at most $n + 1$.

While the longest word in an $\equiv_x$-class is unique, there may be in general more than one shortest word. Consider for

	Property	Conditions
(1.1)	$\frac{f(wv)-E(wv)}{N(wv)} > \frac{f(w)-E(w)}{N(w)}$	$N(wv) < N(w), \quad \rho(wv) \leq \rho(w)$
(1.2)	$\left\lvert\frac{f(wv)-E(wv)}{N(wv)}\right\rvert > \left\lvert\frac{f(w)-E(w)}{N(w)}\right\rvert$	$N(wv) < N(w), \quad \rho(wv) \leq \rho(w)$ and $f(w) > E(w)\frac{\gamma N(w)+N(wv)}{N(w)+N(wv)}$
(1.3)	$f(wv)-E(wv) > f(w)-E(w)$	$E(wv) < E(w)$
(1.4)	$\frac{f(wv)}{E(wv)} > \frac{f(w)}{E(w)}$	$E(wv) < E(w)$
(1.5)	$\frac{f(wv)-E(wv)}{E(wv)} > \frac{f(w)-E(w)}{E(w)}$	$E(wv) < E(w)$
(1.6)	$\frac{f(wv)-E(wv)}{\sqrt{E(wv)}} > \frac{f(w)-E(w)}{\sqrt{E(w)}}$	$E(wv) < E(w)$
(1.7)	$\left\lvert\frac{f(wv)-E(wv)}{\sqrt{E(wv)}}\right\rvert > \left\lvert\frac{f(w)-E(w)}{\sqrt{E(w)}}\right\rvert$	$E(w) > E(wv), \quad f(w) > E(w)\sqrt{\gamma}$
(1.8)	$\frac{(f(wv)-E(wv))^2}{E(wv)} > \frac{(f(w)-E(w))^2}{E(w)}$	$E(w) > E(wv), \quad f(w) > E(w)\sqrt{\gamma}$

Table 1: General monotonicities for scores associated with the counts f, under the hypothesis $f(w) = f(wv)$. We have set $\rho(w) \equiv E(w)/N(w)$ and $\gamma \equiv E(wv)/E(w)$.

	Property	Conditions
(2.1)	$E(Z_{wv}) < E(Z_w)$	none
(2.2)	$f(wv)-E(Z_{wv}) > f(w)-E(Z_w)$	$f(w) = f(wv)$
(2.3)	$\frac{f(wv)}{E(Z_{wv})} > \frac{f(w)}{E(Z_w)}$	$f(w) = f(wv)$
(2.4)	$\frac{f(wv)-E(Z_{wv})}{E(Z_{wv})} > \frac{f(w)-E(Z_w)}{E(Z_w)}$	$f(w) = f(wv)$
(2.5)	$\frac{f(wv)-E(Z_{wv})}{\sqrt{E(Z_{wv})}} > \frac{f(w)-E(Z_w)}{\sqrt{E(Z_w)}}$	$f(w) = f(wv)$
(2.6)	$\left\lvert\frac{f(wv)-E(Z_{wv})}{\sqrt{E(Z_{wv})}}\right\rvert > \left\lvert\frac{f(w)-E(Z_w)}{\sqrt{E(Z_w)}}\right\rvert$	$f(w) = f(wv), \quad f(w) > E(Z_w)\sqrt{\gamma}$
(2.7)	$\frac{(f(wv)-E(Z_{wv}))^2}{E(Z_{wv})} > \frac{(f(w)-E(Z_w))^2}{E(Z_w)}$	$f(w) = f(wv), \quad f(w) > E(Z_w)\sqrt{\gamma}$
(2.8)	$\frac{f(wv)-E(Z_{wv})}{\sqrt{E(Z_{wv})(1-\hat{p}\hat{q})}} > \frac{f(w)-E(Z_w)}{\sqrt{E(Z_w)(1-\hat{p})}}$	$f(w) = f(wv), \quad \hat{p} < 1/2$
(2.9)	$Var(Z_{wv}) < Var(Z_w)$	$p_{max} < 1/\sqrt[m]{4m}$
(2.10)	$\frac{E(Z_{wv})}{\sqrt{Var(Z_{wv})}} < \frac{E(Z_w)}{\sqrt{Var(Z_w)}}$	$p_{max} < \sqrt{2}-1$
(2.11)	$\frac{f(wv)-E(Z_{wv})}{\sqrt{Var(Z_{wv})}} > \frac{f(w)-E(Z_w)}{\sqrt{Var(Z_w)}}$	$f(w) = f(wv), \quad p_{max} < \min\{1/\sqrt[m]{4m}, \sqrt{2}-1\}$
(2.12)	$\left\lvert\frac{f(wv)-E(Z_{wv})}{\sqrt{Var(Z_{wv})}}\right\rvert > \left\lvert\frac{f(w)-E(Z_w)}{\sqrt{Var(Z_w)}}\right\rvert$	$f(w) = f(wv), \quad p_{max} < \min\{1/\sqrt[m]{4m}, \sqrt{2}-1\}$ and $f(w) > E(Z_w)\frac{\gamma\sqrt{Var(Z_w)}+\sqrt{Var(Z_{wv})}}{\sqrt{Var(Z_w)}+\sqrt{Var(Z_{wv})}}$

Table 2: Monotonicities for scores associated with the number of occurrences f under the Bernoulli model for the random variable Z. We set $\gamma \equiv E(Z_{wv})/E(Z_w)$.

	Property	Conditions
(3.1)	$\hat{E}(Z_{wv}) \leq \hat{E}(Z_w)$	none
(3.2)	$f(wv) - \hat{E}(Z_{wv}) \geq f(w) - \hat{E}(Z_w)$	$f(w) = f(wv)$
(3.3)	$\frac{f(wv)}{\hat{E}(Z_{wv})} \geq \frac{f(w)}{\hat{E}(Z_w)}$	$f(w) = f(wv)$
(3.4)	$\frac{f(wv) - \hat{E}(Z_{wv})}{\hat{E}(Z_{wv})} \geq \frac{f(w) - \hat{E}(Z_w)}{\hat{E}(Z_w)}$	$f(w) = f(wv)$
(3.5)	$\frac{f(wv) - \hat{E}(Z_{wv})}{\sqrt{\hat{E}(Z_{wv})}} \geq \frac{f(w) - \hat{E}(Z_w)}{\sqrt{\hat{E}(Z_w)}}$	$f(w) = f(wv)$
(3.6)	$\left\lvert \frac{f(wv) - \hat{E}(Z_{wv})}{\sqrt{\hat{E}(Z_{wv})}} \right\rvert \geq \left\lvert \frac{f(w) - \hat{E}(Z_w)}{\sqrt{\hat{E}(Z_w)}} \right\rvert$	$f(w) = f(wv), \quad f(w) > E(Z_w)\sqrt{\gamma}$
(3.7)	$\frac{(f(wv) - \hat{E}(Z_{wv}))^2}{\hat{E}(Z_{wv})} \geq \frac{(f(w) - \hat{E}(Z_w))^2}{\hat{E}(Z_w)}$	$f(w) = f(wv), \quad f(w) > E(Z_w)\sqrt{\gamma}$

Table 3: Monotonicities for scores associated with the number of occurrences f under Markov model for the random variable Z. We set $\gamma \equiv E(Z_{wv})/E(Z_w)$.

	Property	Conditions
(4.1)	$E(W_{wv}) < E(W_w)$	none
(4.2)	$c(wv) - E(W_{wv}) > c(w) - E(W_w)$	$c(w) = c(wv)$
(4.3)	$\frac{c(wv)}{E(W_{wv})} > \frac{c(w)}{E(W_w)}$	$c(w) = c(wv)$
(4.4)	$\frac{c(wv) - E(W_{wv})}{E(W_{wv})} > \frac{c(w) - E(W_w)}{E(W_w)}$	$c(w) = c(wv)$
(4.5)	$\frac{c(wv) - E(W_{wv})}{\sqrt{E(W_{wv})}} > \frac{c(w) - E(W_w)}{\sqrt{E(W_w)}}$	$c(w) = c(wv)$
(4.6)	$\left\lvert \frac{c(wv) - E(W_{wv})}{\sqrt{E(W_{wv})}} \right\rvert > \left\lvert \frac{c(w) - E(W_w)}{\sqrt{E(W_w)}} \right\rvert$	$c(w) = c(wv), \quad c(w) > E(W_w)\sqrt{\gamma}$
(4.7)	$\frac{(c(wv) - E(W_{wv}))^2}{E(W_{wv})} > \frac{(c(w) - E(W_w))^2}{E(W_w)}$	$c(w) = c(wv), \quad c(w) > E(W_w)\sqrt{\gamma}$

Table 4: Monotonicities of the scores associated with the number of colors c under Bernoulli or Markov model for the random variable W. We set $\gamma \equiv E(W_{wv})/E(W_w)$.

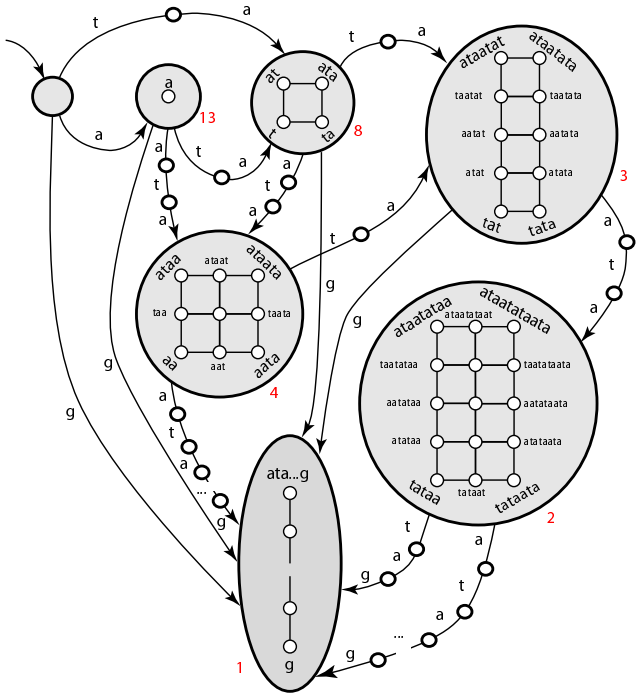

Figure 2: A representation of the seven $\equiv_x$-equivalent classes for $x = \mathtt{ataatataataatataatatag}$. The words in each class can be organized in a lattice. Numbers refer to the number of occurrences

example the text $x = \mathtt{a}^k\mathtt{g}^k$, with $k > 0$. Choosing $k = 2$ yields a class which has three words of length two as minimal elements, namely, **aa**, **gg**, and **ag**. (In fact, $imp_x(\mathtt{aa}) = imp_x(\mathtt{gg}) = imp_x(\mathtt{ag}) = \mathtt{aagg}$.) Taking instead $k = 1$, all three substrings of $x = \mathtt{ag}$ coalesce into a single class which has two shortest words.

Each one of the equivalence classes discussed can be mapped to the nodes of a corresponding automaton or word graph, which becomes thereby the natural support for our statistical tables. The table takes linear space, since the number of classes is linear in $|x|$. The automata themselves are built by classical algorithms or easy adaptations thereof, to be described in more detail in the final version of this extended abstract. The graph for $\equiv_x$, for instance, is built using the fact that an $\equiv_x$-class can be expressed as the union of some left-equivalent classes or, alternatively, as the union of some right-equivalent classes. As the example above shows, however, there are cases in which we *cannot* merge left- or right-equivalent classes without loosing the uniqueness of the shortest word. A consequence of this is that we can use the graph for $\equiv_x$-classes when we are interested in detecting only over-represented words. If under-represented words are also wanted, then we must represent a same $\equiv_x$-class once for each distinct shortest word in it. This results in a substring partition slightly coarser than $\equiv_x$, which will be denoted by $\tilde{\equiv}_x$.

We omit many details and concentrate now on computational results. In [2, 3], linear-time algorithms are given to compute and store expected value $E(Z)$ and variance $Var(Z)$ for the number of occurrences under Bernoulli model of *all* prefixes of a given string. Combining this with the structure of our supporting graph we prove:

THEOREM 4.1. *Under the Bernoulli model, the sets $\mathcal{O}_z^T$ and $\mathcal{U}_z^T$ for Scores*

$$
\begin{aligned}
z_{2.2}(w) &= f(w) - E(Z_w) \\
z_{2.3}(w) &= \frac{f(w)}{E(Z_w)} \\
z_{2.4}(w) &= \frac{f(w) - E(Z_w)}{E(Z_w)} \\
z_{2.5}(w) &= \frac{f(w) - E(Z_w)}{\sqrt{E(Z_w)}} \\
z_{2.8}(w) &= \frac{f(w) - E(Z_w)}{\sqrt{\hat{V}ar(Z_w)}} \quad (\hat{p} < 1/2) \\
z_{2.11}(w) &= \frac{f(w) - E(Z_w)}{\sqrt{Var(Z_w)}} \quad (p_{max} < \min\{\frac{1}{\sqrt[m]{4m}}, \sqrt{2} - 1\})
\end{aligned}
$$

and the set $\mathcal{S}_z^T$ for Scores

$$
\begin{aligned}
z_{2.6}(w) &= \left| \frac{f(w) - E(Z_w)}{\sqrt{E(Z_w)}} \right| \\
z_{2.7}(w) &= \frac{(f(w) - E(Z_w))^2}{E(Z_w)} \\
z_{2.12}(w) &= \left| \frac{f(w) - E(Z_w)}{\sqrt{Var(Z_w)}} \right| \quad (p_{max} < \min\{\frac{1}{\sqrt[m]{4m}}, \sqrt{2} - 1\})
\end{aligned}
$$

can be computed in linear time and space.

The computation of $E(Z)$ in Markov models is more difficult than with Bernoulli. Recall the maximum likelihood estimator for the expectation in Equation 1. If we compute the (Markov) prefix product $pp(i)$ as follows

$$
pp(i) = \begin{cases} 1 & \text{if } i = 0 \\ \prod_{j=1}^{i} \frac{f(x_{[j,j+M]})}{f(x_{[j,j+M-1]})} & \text{if } 1 \le i \le n \end{cases}
$$

then $\hat{E}(Z_y)$ is rewrittten as

$$
\hat{E}(Z_y) = f(y_{[1,M+1]}) \, \frac{pp(e - M)}{pp(b)}
$$

where (b, e) are the beginning and the ending position of any of the occurrences of y in x. Hence, if $f(y_{[1,M+1]})$ and the vector $pp(i)$ are available, we can compute $\hat{E}(Z_y)$ in constant time.

It is not difficult to compute the products $pp(i)$ of interest in overall linear time. When working with multisequences, we have to build a vector of prefix products for each sequence using the global statistics of occurrences of each word of size M and $M + 1$. We also build the Bernoulli prefix products to compute $E(Z)$ for words smaller than $M + 2$, because the estimator of $\hat{E}(Z)$ cannot be used for these words. The resulting algorithm is linear in the total size of the multisequence.

The following theorem summarizes the results obtained.

THEOREM 4.2. *Under Markov models, the sets $\mathcal{O}_z^T$ and $\mathcal{U}_z^T$ for Scores*

$$\begin{aligned} z_{3.2}(w) &= f(w) - \hat{E}(Z_w) \\ z_{3.3}(w) &= \frac{f(w)}{\hat{E}(Z_w)} \\ z_{3.4}(w) &= \frac{f(w) - \hat{E}(Z_w)}{\hat{E}(Z_w)} \\ z_{3.5}(w) &= \frac{f(w) - \hat{E}(Z_w)}{\sqrt{\hat{E}(Z_w)}} \end{aligned}$$

and the set $\mathcal{S}_z^T$ for Scores

$$\begin{aligned} z_{3.6}(w) &= \left| \frac{f(w) - \hat{E}(Z_w)}{\sqrt{\hat{E}(Z_w)}} \right| \\ z_{3.7}(w) &= \frac{(f(w) - \hat{E}(Z_w))^2}{\hat{E}(Z_w)} \end{aligned}$$

can be computed in linear time and space.

We now turn to color counts in multisequences. The computation of $E(W)$ and $Var(W)$ can be accomplished once array $\{E(Z_y^j) : j \in [1 \dots k]\}$, that is, the expected number of occurrences of y in each sequence is available. $E(Z_y^j)$ has to be evaluated on the local model estimated *only* from the j-th sequence. Once that all $E(Z_y^j)$ are available we can use Equation 2 to compute $E(W_y)$ and $Var(W_y)$.

Having k different sets of parameters to handle makes the usage of the prefix products slightly more involved. For any word y, we have to estimate its expected number of occurrences in *each* sequence, even in sequences in which y does not appear at all. Therefore, we cannot compute only *one* prefix product for each sequence. We need to compute k vectors of prefix products for each sequence at an overall $O(kn)$ time- and space complexity for the preprocessing phase, where we assume $n = \sum_{i=1}^{k} \left|x^{(i)}\right|$. We need an additional vector in which we record the starting position of any of the occurrences of y in each sequence. The resulting algorithm has overall time complexity $O(kn)$.

The following theorem summarizes this discussion.

THEOREM 4.3. *Under Bernoulli or Markov models, the sets $\mathcal{O}_z^T$ and $\mathcal{U}_z^T$ of a multisequence $\{x^{(1)}, x^{(2)}, \dots, x^{(k)}\}$ for Scores*

$$\begin{aligned} z_{4.2}(w) &= c(w) - E(W_w) \\ z_{4.3}(w) &= \frac{c(w)}{E(W_w)} \\ z_{4.4}(w) &= \frac{c(w) - E(W_w)}{E(W_w)} \\ z_{4.5}(w) &= \frac{c(w) - E(W_w)}{\sqrt{E(W_w)}} \end{aligned}$$

and the set $\mathcal{S}_z^T$ for Scores

$$\begin{aligned} z_{4.6}(w) &= \left| \frac{c(w) - E(W_w)}{\sqrt{E(W_w)}} \right| \\ z_{4.7}(w) &= \frac{(c(w) - E(W_w))^2}{(W_w)} \end{aligned}$$

can be computed in $O(k \sum_{i=1}^{k} \left|x^{(i)}\right|)$ time and space.

5. CONCLUSIONS

We have shown that under several scores and models, we can bound the number of candidate over- and under-represented words in a sequence and carry out the related computations in correspondingly efficient time and space. Our results require that the scores under consideration grow monotonically for words in each class of a partition of which the index or number of classes is linear in the textstring. As seen in this paper, such a condition is met by many scores. The corresponding statistical tables take up the form of some variant of a trie structure of which the branching nodes, in a number linear in the textstring length, are all and only the sites where a score needs be computed and displayed. In practice, additional space savings could achieved by grouping in a same equivalence class consecutive branching nodes in a chain of nodes in which the scores are non-decreasing. For instance, this could be based on the condition that the difference of observed and expected frequency is larger for the longer word and the normalization term is decreasing for the longer word. (The case of fixed frequency for both words is just a special case of this.) Note that in such a variant of the trie the words in an equivalence class are no longer characterized by having essentially the same list of occurrences. Another way of giving the condition is to say that the ratio of the frequency of the longer word to that of the shorter word should be larger than the ratio of their corresponding expectations. In this case, the longer word has the bigger score. Still, an important question regards more the generation of tables for general scores, particularly for those that do not necessarily meet those monotonicity conditions. There are two qualifications to the problem, respectively regarding space and construction time. As far as space is concerned, we have seen that the crucial handle towards linear space is represented by equivalence class partitions $\{C_1, C_2, \dots, C_l\}$ that satisfy properties such as at the beginning of Section 4. Clearly, the equivalence relations $\equiv_l$, $\equiv_r$ and $\equiv_x$ all meet these conditions. We note that a class C_i in any of the corresponding partitions represents a maximal set of strings that occur precisely at the same positions in x, possibly up to some small uniform offset. For our purposes, any such class may be fully represented by the quadruplet $\{\max(C_i), \min(C_i), (i_1, l_1, z_{max}), (i_2, l_2, z_{min})\}$ where (i_1, l_1, z_{max}) and (i_2, l_2, z_{min}) give the positions, lengths and scores of the substrings of $\max(C_i)$ achieving the largest and smallest score values, respectively. The monotonicity conditions studied in this paper automatically assign z_{max} to $\max(C_i)$ and z_{min} to $\min(C_i)$, thereby rendering redundant the position information in a quadruplet. In addition, when dealing with $\equiv_l$ (respectively, $\equiv_r$), we also know that $\min(C_i)$ is a prefix (resp., suffix) of $\max(C_i)$, which brings even more savings. In the general case, a linear number of quadruplets such as above fully characterizes the set of unusual words. This is true, in particular, for the partition associ-

ated with the equivalence relation $\hat{\equiv}_x$, which achieves the smallest number of classes under the constrains of Section 4. The corresponding graph may thus serve as the natural support of exhaustive statistical tables for the most general models. The computational costs involved in producing such tables might pose further interesting problems of algorithm design.

Acnowledgements

The passage by J.L. Borges which inspired the title of [1] was pointed out to the author by Gustavo Stolovitzky. We are also grateful to the Referees for their helpful comments.

6. REFERENCES

[1] Apostolico, A. Of maps bigger than the empire. In *Keynote, Proceedings of the 8th International Colloquium on String Processing and Information Retrieval* (Laguna de San Rafael, Chile, November 2001), IEEE Computer Society Press, pp. 2–10.

[2] Apostolico, A., Bock, M. E., Lonardi, S., and Xu, X. Efficient detection of unusual words. *J. Comput. Bio. 7*, 1/2 (Jan. 2000), 71–94.

[3] Apostolico, A., Bock, M. E., and Xu, X. Annotated statistical indices for sequence analysis. In *Compression and Complexity of Sequences* (Positano, Italy, 1998), B. Carpentieri, A. De Santis, U. Vaccaro, and J. Storer, Eds., IEEE Computer Society Press, pp. 215–229.

[4] Apostolico, A., and Galil, Z., Eds. *Pattern matching algorithms*. Oxford University Press, New York, NY, 1997.

[5] Apostolico, A., and Lonardi, S. Verbumculus. `http://www.cs.ucr.edu/~stelo/Verbumculus`, 2001.

[6] Blumer, A., Blumer, J., Ehrenfeucht, A., Haussler, D., and McConnel, R. Complete inverted files for efficient text retrieval and analysis. *J. Assoc. Comput. Mach. 34*, 3 (1987), 578–595.

[7] Borges, J. L. *A Universal History of Infamy*. Penguin Books, London, 1975.

[8] Clift, B., Haussler, D., McConnell, R., Schneider, T. D., and Stormo, G. D. Sequences landscapes. *Nucleic Acids Res. 14* (1986), 141–158.

[9] Gentleman, J. The distribution of the frequency of subsequences in alphabetic sequences, as exemplified by deoxyribonucleic acid. *Appl. Statist. 43* (1994), 404–414.

[10] Kleffe, J., and Borodovsky, M. First and second moment of counts of words in random texts generated by Markov chains. *Comput. Appl. Biosci. 8* (1992), 433–441.

[11] Leung, M. Y., Marsh, G. M., and Speed, T. P. Over and underrepresentation of short DNA words in herpesvirus genomes. *J. Comput. Bio. 3* (1996), 345–360.

[12] Lonardi, S. *Global Detectors of Unusual Words: Design, Implementation, and Applications to Pattern Discovery in Biosequences*. PhD thesis, Department of Computer Sciences, Purdue University, August 2001.

[13] Lundstrom, R. *Stochastic models and statistical methods for DNA sequence data*. PhD thesis, University of Utah, 1990.

[14] Pevzner, P. A., Borodovsky, M. Y., and Mironov, A. A. Linguistics of nucleotides sequences I: The significance of deviations from mean statistical characteristics and prediction of the frequencies of occurrence of words. *J. Biomol. Struct. Dynamics 6* (1989), 1013–1026.

[15] Régnier, M., and Szpankowski, W. On pattern frequency occurrences in a Markovian sequence. *Algorithmica 22* (1998), 631–649.

[16] Reinert, G., Schbath, S., and Waterman, M. S. Probabilistic and statistical properties of words: An overview. *J. Comput. Bio. 7* (2000), 1–46.

[17] Stűckle, E., Emmrich, C., Grob, U., and Nielsen, P. Statistical analysis of nucleotide sequences. *Nucleic Acids Res. 18*, 22 (1990), 6641–6647.

[18] Waterman, M. S. *Introduction to Computational Biology*. Chapman & Hall, 1995.