JOURNAL OF COMPUTATIONAL BIOLOGY
Volume 10, Numbers 3–4, 2003

Pp. 283–311

Monotony of Surprise and Large-Scale Quest for Unusual Words

ALBERTO APOSTOLICO,[1] MARY ELLEN BOCK,[2] and STEFANO LONARDI[3]

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 subword 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.

Key words: design and analysis of algorithms, combinatoric on words, statistical analysis of sequences, annotated suffix trees, over- and under-represented words, pattern discovery.

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., Reinert *et al.* [2000], Waterman [1995]). 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

[1]Department of Computer Sciences, Purdue University, West Lafayette, IN 47907, and Dipartimento di Ingegneria dell'Informazione, Università di Padova, Padova, Italy.
[2]Department of Statistics, Purdue University, West Lafayette, IN 47907.
[3]Department of Computer Science and Engineering, University of California, Riverside, CA 92521.

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 (Apostolico, 2001; Borges, 1975): "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]

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 measure 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, overrepresented 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 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 (nontrivial) 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 requirement and are thus susceptible to this treatment. The main results come in the form of a series of conditions and properties, which we describe here within a framework 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 Section 3. Finally, we briefly discuss the algorithmic implications and constructs in Section 4. We also highlight future work, and extend succinct descriptors of the kind considered here to more general models and areas outside of the monotonicity realm. These results are

[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 Adolfo Bioy Casares. English translation quoted from Borges (1975): "... 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."

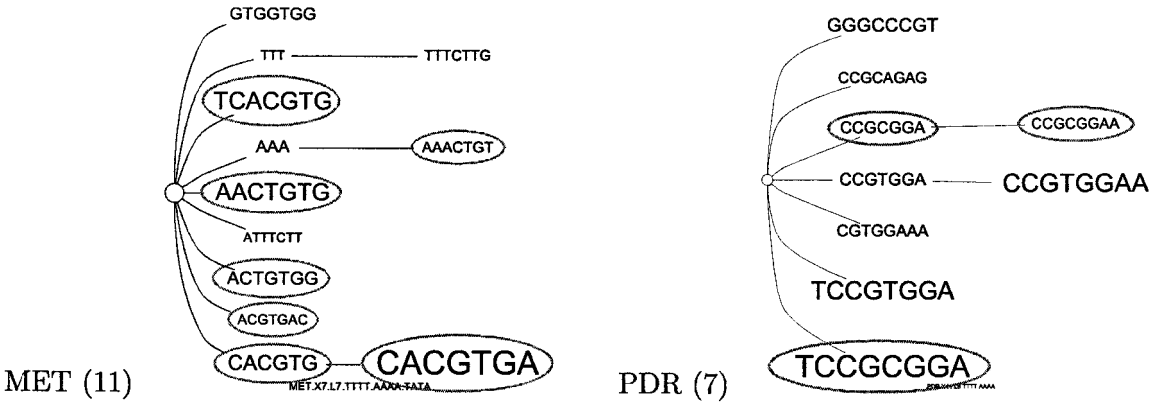

FIG. 1. Overrepresented words in a set of coregulated genes. A word's increasing departure from its expected frequency is rendered by proportionally increased font size. Superposition of the words circled by hand yields the previously known motifs: TCACGTG and AAAACTGTGG in the MET family of 11 sequences,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

being incorporated into an existing suite of programs (Lonardi, 2001; Apostolico and Lonardi, 2001). As an example demonstration, Fig. 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 gene expression in relation to development, tissue specificity, and/or environment. Each one of the two families contains a set of coregulated 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 overrepresented across the family. In this, like in 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, testing, and fine tuning of hypotheses.

2. PRELIMINARIES

We use standard concepts and notation about strings, for which we refer to (Apostolico *et al.*, 1998, 2000; Apostolico and Galil, 1997). For a substring y of a text x, we denote by $f(y)$ the number of occurrences of y in x. We have $f(y) = |pos_x(y)| = |endpos_x(y)|$, where $pos_x(y)$ is the *start-set* of starting positions of y in x and $endpos_x(y)$ is the similarly defined *end-set*. 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 ergodic 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,[2] have been given by several authors (see, e.g., Pevzner *et al.* [1989], Stückle *et al.* [1990], Kleffe and

[2]Although "multinomial" would be the appropriate term for larger than binary alphabets, we conform here to the current usage and adopt the word "Bernoulli" throughout.

Borodovsky [1992], Gentleman [1994], Régnier and Szpankowski [1998]). Here we adopt derivations in Apostolico *et al.* (1998, 2000). With p_a the probability of symbol $a \in \Sigma$ and $\hat{p} = \prod_{i=1}^{m} p_{y_{[i]}}$, we have

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

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

where

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

is the *auto-correlation factor* of y that depends on the set $\mathcal{P}(y)$ of the lengths of the periods[3] of y. In cases of practical interest, we expect $m \le (n+1)/2$, 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 (Reinert *et al.*, 2000):

$$\hat{E}(Z_y) = \frac{\prod_{i=1}^{m-M} f(y_{[i,i+M]})}{\prod_{i=2}^{m-M} f(y_{[i,i+M-1]})} = f(y_{[1,M+1]}) \prod_{j=2}^{m-M} \frac{f(y_{[j,j+M]})}{f(y_{[j,j+M-1]})}. \tag{2}$$

The expression for the variance $Var(Z_y)$ for Markov chains is very involved. Complete derivations have been given by Lundstrom (1990), Kleffe and Borodovsky (1992), and Régnier and Szpankowski (1998). 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., Reinert *et al.* [2000]). 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 (1990) and is clearly different from the asymptotic variance of $E(Z_y)$ (see Waterman [1995] 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_y for the number of colors can be computed by

$$E(W_y) = k - \sum_{j=1}^{k} \mathbf{P}[Z_y^j = 0] \tag{3}$$

because $E(W_j) = \mathbf{P}[Z_y^j \neq 0]$.

[3]String z has a *period* w if z is a nonempty prefix of w^k for some integer $k \ge 1$.

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 (Leung *et al.*, 1996). For any word w, a standardized frequency called the *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.

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 overcome 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-overrepresented* in C (resp., *T-underrepresented*) 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.

Now let 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-overrepresented 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-underrepresented words of C_i, and $\mathcal{S}_z^T$ the set of all T-surprising words, $1 \leq i \leq l$.

For two strings u and $v = suz$, 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., *nondecreasing*) 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)-*nonincreasing* 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 overrepresented from underrepresented words. This restriction is compensated by the fact that we can now relax the property asked of the score function, as will be 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 convex 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 concave 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

This section displays a collection of monotonicity results established with regard to the models and z-scores considered.

Recall that we consider score functions of the form

$$z(w) = \frac{f(w) - E(w)}{N(w)}$$

where $f(w) > 0$, $E(w) > 0$, and $N(w) > 0$ where $N(w)$ appears in the score as the expected value of some function of w.

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 $\rho(w) \equiv E(w)/N(w)$. First, we state a simple fact on the monotonicity of $E(w)$ given the monotonicity of $\rho(w)$ and $N(w)$.

Fact 3.1. *If $\rho(w) \geq \rho(wv)$, and if $N(w) > N(wv)$, then $E(w) > E(wv)$.*

Proof. From $\rho(w) \geq \rho(wv)$, we get that $E(w)/E(wv) \geq N(w)/N(wv)$. By hypothesis, $N(w)/N(wv) > 1$, whence the claim. ■

Under some general conditions on $N(w)$ and $\rho(w)$, we can prove the monotonicity of any score functions of the form described above.

Theorem 3.1. *If $f(w) = f(wv)$, $N(wv) < N(w)$, and $\rho(wv) \leq \rho(w)$, then*

$$\frac{f(wv) - E(wv)}{N(wv)} > \frac{f(w) - E(w)}{N(w)}.$$

Proof. By construction of the equivalence classes, we have $f(wv) = f(w) \geq 0$. We can rewrite the inequality of the theorem as

$$\frac{f(w)}{E(wv)}\left(1 - \frac{N(wv)}{N(w)}\right) > 1 - \frac{\rho(w)}{\rho(wv)}.$$

The left hand side is always positive because $0 < N(wv)/N(w) < 1$, and the right hand size is always negative (or zero if $\rho(w) = \rho(wv)$). ■

The statement of Theorem 3.1 also holds by exchanging the condition $\rho(wv) \leq \rho(w)$ with $f(w) > E(w) > E(wv)$. Let us now apply the theorem to some common choices for $N(w)$.

Fact 3.2. *If $f(w) = f(wv)$ and $E(wv) < E(w)$, then*

1. $f(wv) - E(wv) > f(w) - E(w)$,
2. $\dfrac{f(wv)}{E(wv)} > \dfrac{f(w)}{E(w)}$,
3. $\dfrac{f(wv) - E(wv)}{E(wv)} > \dfrac{f(w) - E(w)}{E(w)}$,
4. $\dfrac{f(wv) - E(wv)}{\sqrt{E(wv)}} > \dfrac{f(w) - E(w)}{\sqrt{E(w)}}$.

Proof.

1. The choice $N(w) = 1$, $\rho(w) = E(w)$ satisfies the conditions of Theorem 3.1 because $E(wv) < E(w)$;
2. by hypothesis $0 < 1/E(w) < 1/E(wv)$, and we have that $f(w) = f(wv)$;
3. the choice $N(w) = E(w)$, $\rho(w) = 1$ satisfies the conditions of Theorem 3.1 because $E(wv) < E(w)$;
4. the choice $N(w) = \sqrt{E(w)}$, $\rho(w) = \sqrt{E(w)}$ satisfies the conditions of Theorem 3.1 because $E(wv) < E(w)$. ■

Other types of scores use absolute values or powers of the difference $f - E$.

Theorem 3.2. *If $f(w) = f(wv) \equiv f$, $N(wv) < N(w)$, and $\rho(wv) \leq \rho(w)$, then*

$$\left|\frac{f(wv) - E(wv)}{N(wv)}\right| > \left|\frac{f(w) - E(w)}{N(w)}\right| \quad \text{iff} \quad f > E(w)\frac{\gamma N(w) + N(wv)}{N(w) + N(wv)}$$

where $\gamma = E(wv)/E(w)$.

Proof. Note first that $0 < \gamma < 1$ by Fact 3.1 and that

$$E(wv) = E(w)\gamma < E(w)\frac{\gamma N(w) + N(wv)}{N(w) + N(wv)} < E(w).$$

We set, for convenience, $E^* = E(w)\frac{\gamma N(w)+N(wv)}{N(w)+N(wv)}$.

We first prove that if $f > E^*$ then $|z(wv)| > |z(w)|$. We consider two cases, one of which is trivial. When $f > E(w)$, then both $f(wv) - E(wv)$ and $f(w) - E(w)$ are positive and the claim follows directly from Fact 3.2. If instead $E^* < f < E(w)$, we evaluate the difference of the scores

$$\begin{aligned} N(wv)N(w)\,(|z(wv)| - |z(w)|) &= N(wv)N(w)\left(\frac{f - \gamma E(w)}{N(wv)} + \frac{f - E(w)}{N(w)}\right) \\ &= (f - \gamma E(w))N(w) + (f - E(w))N(wv) \\ &= f(N(w) + N(wv)) - E(w)(\gamma N(w) + N(wv)) \\ &= (N(w) + N(wv))(f - E^*) \end{aligned}$$

which is positive by hypothesis.

The converse can be proved by showing that if $f \leq E^*$ we have $|z(wv)| \leq |z(w)|$. Again, there are two cases, one of which is trivial. When $0 < f(w) < E(wv)$, both $f(wv) - E(wv)$ and $f(w) - E(w)$ are negative and the claim follows directly from Fact 3.2. If instead $E(wv) < f \leq E^*$, we use the relation obtained above, i.e.,

$$|z(wv)| - |z(w)| = \frac{N(w) + N(wv)}{N(wv)N(w)}(f - E^*),$$

to get the claim. ■

Theorem 3.2 say that these scores are monotonically decreasing when $f < E^*$ and monotonically increasing when $f > E^*$. We can picture the dynamics of the score as follows. 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 begins to grow.

Fact 3.3. *If $f(w) = f(wv)$ and if $E(w) > E(wv) \equiv \gamma\, E(w)$, then*

1. $\left|\frac{f(wv) - E(wv)}{\sqrt{E(wv)}}\right| > \left|\frac{f(w) - E(w)}{\sqrt{E(w)}}\right| \quad \text{iff} \quad f(wv) > E(w)\sqrt{\gamma},$

2. $\frac{(f(wv) - E(wv))^2}{E(wv)} > \frac{(f(w) - E(w))^2}{E(w)} \quad \text{iff} \quad f(wv) > E(w)\sqrt{\gamma}.$

Proof. Relation (1) follows directly from Theorem 3.2 by setting $N(w) = \sqrt{E(w)}$. Relation (2) follows from relation (1) by squaring both sides. ■

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.

Fact 3.4. *Given a threshold $T > 0$ on the number of occurrences, then*

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

Proof. From $f(uwv) \leq f(w)$ we know that if $f(w) \leq T$ then also $f(wv) \leq T$. Therefore $\mathbf{P}(f(w) \leq T) \leq \mathbf{P}(f(wv) \leq T)$. ■

Let us consider the score

$$\begin{aligned} z_P(w, T) &= \min\{\mathbf{P}(f(w) \leq T), \mathbf{P}(f(w) > T)\} \\ &= \min\{\mathbf{P}(f(w) \leq T), 1 - \mathbf{P}(f(w) \leq T)\} \end{aligned}$$

evaluated on the strings in a class C. By Fact 3.4, one can compute the score only for the shortest and the longest strings in C, as follows:

$$\min\{\mathbf{P}(f(\min(C)) \leq T), \mathbf{P}(f(\max(C)) > T)\}.$$

Also, note that score $z_P(w, T)$ satisfies the conditions of Fact 2.3. In fact, $z' = \mathbf{P}(f(w) \leq T)$ is $(min(C), max(C))$-monotonic by Fact 3.4, and the transformation $z = \min\{z', 1-z'\}$ is a concave function in z'.

Table 1 summarizes the collection of these properties.

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

3.1. The expected number of occurrences under Bernoulli

Let p_a be 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 $p_{max} = \max_{a\in\Sigma} p_a$. We also observe that $p_{max} \geq 1/|\Sigma|$, and therefore upper bounds on p_{max} could turn out to be unsatisfiable for small alphabets.

Fact 3.5. *Let x be a text generated by a Bernoulli process. Then $E(Z_{wv}) < E(Z_w)$.*

Proof. We have

$$\frac{E(Z_{wv})}{E(Z_w)} = \frac{(n-|w|-|v|+1)\hat{p}\hat{q}}{(n-|w|+1)\hat{p}} = \left(1 - \frac{|v|}{n-|w|+1}\right)\hat{q} < \hat{q} < 1$$

because $\frac{|v|}{n-|w|+1} > 0$. ■

Fact 3.6. *Let x be a text generated by a Bernoulli process. If $f(w) = f(wv)$, then*

1. $f(wv) - E(Z_{wv}) > f(w) - E(Z_w)$,
2. $\dfrac{f(wv)}{E(Z_{wv})} > \dfrac{f(w)}{E(Z_w)}$,
3. $\dfrac{f(wv) - E(Z_{wv})}{E(Z_{wv})} > \dfrac{f(w) - E(Z_w)}{E(Z_w)}$,
4. $\dfrac{f(wv) - E(Z_{wv})}{\sqrt{E(Z_{wv})}} > \dfrac{f(w) - E(Z_w)}{\sqrt{E(Z_w)}}$.

Proof. Directly from Theorem 3.1 and Fact 3.5. ■

Fact 3.7. *Let x be a text generated by a Bernoulli process. If $f(w) = f(wv) \equiv f$, then*

1. $\left|\dfrac{f(wv) - E(Z_{wv})}{\sqrt{E(Z_{wv})}}\right| > \left|\dfrac{f(w) - E(Z_w)}{\sqrt{E(Z_w)}}\right|$ *iff* $f > E(Z_w)\sqrt{\gamma}$,
2. $\dfrac{(f(wv) - E(Z_{wv}))^2}{E(Z_{wv})} > \dfrac{(f(w) - E(Z_w))^2}{E(Z_w)}$ *iff* $f > E(Z_w)\sqrt{\gamma}$

where $\gamma = E(Z_{wv})/E(Z_w)$.

Proof. Directly from Fact 3.3 and Fact 3.5. ■

A score that is not captured in Fact 3.2 uses the square root of the first order approximation of the variance as the normalizing factor.

Fact 3.8. *Let x be a text generated by a Bernoulli process. If $f(w) = f(wv)$ and $\hat{p} < 1/2$, then*

$$\frac{f(wv) - E(Z_{wv})}{\sqrt{E(Z_{wv})(1-\hat{p}\hat{q})}} > \frac{f(w) - E(Z_w)}{\sqrt{E(Z_w)(1-\hat{p})}}.$$

Proof. To have monotonicity, the functions $N(w) = \sqrt{E(Z_w)(1-\hat{p})}$ and $\rho(w) = E(Z_w)/N(w)$ should satisfy the conditions of Theorem 3.1. First we study the ratio

$$\left(\frac{N(wv)}{N(w)}\right)^2 = \left(1 - \frac{|v|}{n-|w|+1}\right)\frac{\hat{p}\hat{q}(1-\hat{p}\hat{q})}{\hat{p}(1-\hat{p})} < \frac{\hat{p}\hat{q}(1-\hat{p}\hat{q})}{\hat{p}(1-\hat{p})}.$$

The concave product $\hat{p}(1-\hat{p})$ reaches its maximum for $\hat{p} = 1/2$. Since we assume $\hat{p} < 1/2$, the rightmost term is smaller than one. The monotonicity of $N(w)$ is satisfied.

Then, we need to prove that $\rho(w)$ also is monotonic, i.e., $\rho(wv) \leq \rho(w)$, which is equivalent to

$$\frac{E(Z_{wv})}{E(Z_w)} \frac{1-\hat{p}}{1-\hat{p}\hat{q}} \leq 1$$

but $E(Z_{wv})/E(Z_w) < 1$ by hypothesis and $(1-\hat{p})/(1-\hat{p}\hat{q}) < 1$ for any choice of $\hat{p}, \hat{q} \in [0, 1]$. ■

To study the monotonicity of the score with the complete variance, we first must prove some facts about the auto-correlation function

$$B(w) = \sum_{d \in \mathcal{P}(w)} (n - |w| + 1 - d) \prod_{j=|w|-d+1}^{|w|} p_{w_{[j]}}$$

where $\mathcal{P}(w)$ is the set of the period lengths of w. Throughout this section, unless otherwise noted, a is any of the symbols in Σ such that $p_a = p_{max}$.

Fact 3.9. *Let n be the size of a text generated by a Bernoulli process and $2 \leq m \leq (n+1)/2$. If $p_a < (\sqrt{5}-1)/2$, then $p_a^m B(a^m)$ is monotonically decreasing with m.*

Proof. Words a^m have period set $\{1, 2, \ldots, m-1\}$ and, therefore,

$$\begin{aligned}
B(a^m) &= \sum_{l=1}^{m-1} (n-m+1-l) p_a^l = \sum_{k=0}^{m-2} (n-m-k) p_a^{k+1} \\
&= (n-m) p_a \sum_{k=0}^{m-2} p_a^k - p_a \sum_{k=0}^{m-2} k p_a^k \\
&= p_a \left((n-m) \frac{1-p_a^{m-1}}{1-p_a} - \frac{(m-2)p_a^m - (m-1)p_a^{m-1} + p_a}{(1-p_a)^2} \right) \\
&= \frac{p_a}{(1-p_a)^2} \left((n-m)(1-p_a)(1-p_a^{m-1}) - (m-2)p_a^m + (m-1)p_a^{m-1} - p_a \right) \\
&= \frac{p_a}{(1-p_a)^2} \left((n-m)(1-p_a-p_a^{m-1}+p_a^m) - (m-2)p_a^m + (m-1)p_a^{m-1} - p_a \right) \\
&= \frac{p_a}{(1-p_a)^2} \left(n-m-(n-m+1)p_a - (n-2m+1)p_a^{m-1} + (n-2m+2)p_a^m \right).
\end{aligned}$$

We now consider the function $b(m) = p_a^m B(a^m)$ in the interval $n > 0, m \in [2, (n+1)/2], p_a \in (0, 1)$. Since function $b(m)$ is defined for integer values of m, we study the differences between consecutive values of m. We define the function

$$\Delta(m) \equiv \frac{b(m-1) - b(m)}{p_a^m},$$

and after some algebraic manipulations we get

$$\Delta(m) = \frac{B(\alpha_{m-1})}{p_a} - B(a^m) = -p_a^m(n-2m) - p_a^{m-1}(n-2m+1) + (n-m).$$

We first aim our efforts towards small values of m. Specifically, we look for values of p_a and n such that $b(2) - b(3) > 0$. We have

$$\Delta(2) = \frac{b(2) - b(3)}{p_a^3} = -p_a^2(n-4) - p_a(n-3) + (n-2).$$

The solution of the inequality $b(2) - b(3) > 0$ is $0 < p_a < (3 - n + \sqrt{5n^2 - 30n + 41})/(2n - 8)$. This interval shrinks as n grows. Taking the limit $n \to \infty$, we get $0 < p_a < (\sqrt{5} - 1)/2 \approx 0.618$.

Repeating the analysis on $b(3) - b(4)$, we get

$$\Delta(3) = \frac{b(3) - b(4)}{p_a^4} = -p_a^3(n-6) - p_a^2(n-5) + (n-3),$$

which has two imaginary roots and one positive real root. The function is positive in the interval $(0, (C^2 - 2C + 4)/(6C))$ where $C = 100 + 12\sqrt{69}$. The upper extreme of the interval is about 0.7548784213, which is bigger than $(\sqrt{5} - 1)/2$.

As we increase m, the difference $b(m) - b(m+1)$ remains positive for larger and larger intervals. Finally, when $m = (n-1)/2$, we get

$$\Delta\left(\frac{n-1}{2}\right) = \frac{b((n-1)/2) - b((n+1)/2)}{p_a^{(n+1)/2}} = \frac{n+1}{2} - p_a^{(n-3)/2}(2 + p_a).$$

The latter function is *always* positive for any choice of p_a and $n > 5$. In fact, if $n > 5$,

$$\Delta\left(\frac{n-1}{2}\right) = \frac{n+1}{2} - p_a^{(n-3)/2}(2 + p_a) \geq \frac{n+1}{2} - 3 > 0.$$

We can conclude that the most restrictive case is $m = 2$. If we choose $p_a < (\sqrt{5} - 1)/2$, then $b(m)$ is monotonically decreasing when $2 \leq m \leq (n+1)/2$, for any choice of $n > 0$. ■

Fact 3.10. *Let n be the size of a text generated by a Bernoulli process and $2 \leq m \leq (n+1)/2$. For all words $w \in \Sigma^m$, we have*

$$0 \leq B(w) \leq B(a^m) \leq \frac{p_a}{1 - p_a}(n - m) - \frac{p_a^2(1 - p_a^{m-1})}{(1 - p_a)^2}.$$

Proof. We have

$$\begin{aligned}
B(w) &= \sum_{d \in \mathcal{P}(w)} (n - m + 1 - d) \prod_{j=m-d+1}^{m} p_{w_{[j]}} \\
&\leq \sum_{d \in \mathcal{P}(w)} (n - m + 1 - d) p_a^d \\
&\leq \sum_{d \in \mathcal{P}(a^m)} (n - m + 1 - d) p_a^d \\
&= \sum_{d=1}^{m-1} (n - m + 1 - d) p_a^d \\
&= B(a^m)
\end{aligned}$$

since (1) all terms in the sum are positive ($1 \leq d \leq m - 1$ and $m \leq (n+1)/2$), (2) a^m has at least all the periods of w (i.e., $\mathcal{P}(w) \subseteq \mathcal{P}(a^m) = \{1, 2, \ldots, m-1\}$), and (3) $\prod_{j=m-d+1}^{m} p_{w_{[j]}} \leq p_a^d = p_{max}^d$.

From the derivation of $B(a^m)$ in Fact 3.9, we have

$$\begin{aligned}
B(a^m) &= \frac{p_a}{(1-p_a)^2}\left(n-m-(n-m+1)p_a-(n-2m+1)p_a^{m-1}+(n-2m+2)p_a^m\right)\\
&= \frac{p_a}{(1-p_a)^2}\left(n-m-(n-m+1)p_a+p_a^m+p_a^{m-1}(p_a-1)(n-2m+1)\right)\\
&\le \frac{p_a}{(1-p_a)^2}\left(n-m-(n-m+1)p_a+p_a^m\right)\\
&= \frac{p_a}{1-p_a}\left(n-m-\sum_{i=1}^{m-1}p_a^i\right)\\
&= \frac{p_a}{1-p_a}(n-m)-\frac{p_a^2(1-p_a^{m-1})}{(1-p_a)^2}
\end{aligned}$$

because $n-2m+1>0$ and $p_a-1\le 0$. ■

We can now get a simple bound on the maximum value achieved by $\hat{p}B(w)$ for any word $w\in\Sigma^+$.

Corollary 3.1. *Let w be any substring of a text generated by a Bernoulli process, $m=|w|\ge 2$, and a be the symbol in Σ such that $p_a=p_{max}<(\sqrt{5}-1)/2$. Then*

$$0\le \hat{p}B(w)\le (n-2)p_{max}^3.$$

Proof. We already know that $\hat{p}\le p_a^m$, and therefore $\hat{p}B(w)\le p_a^m B(w)$. Fact 3.10 says that $B(a^m)$ is an upper bound for $B(w)$ for any word w of the same length and that $p_a^m B(a^m)$ reach the maximum for $m=2$. Specifically, the maximum is $p_{max}^2 B(\alpha_2)=p_{max}^2(n-m)p_{max}$. ■

We are now ready to study the monotonicity of the score with the "exact" variance. We will warm up studying the family of words a^m.

Fact 3.11. *Let $2\le m\le (n+1)/2$. If $p_a\le 0.6$, then $Var(Z_{a^m})$ is monotonically decreasing with m.*

Proof. We study the function

$$Var(Z_{a^m})=(n-m+1)p_a^m(1-p_a^m)-p_a^{2m}(2n-3m+2)(m-1)+2p_a^m B(a^m)$$

defined on integer values of m. We study the differences between consecutive values of m. We define the function

$$\Delta(m)\equiv\frac{Var(Z_{a^m})-Var(Z_{a^{m+1}})}{p_a^m}.$$

After some algebraic manipulations, we get

$$\begin{aligned}
\Delta(m) = {} & p_a^{m+2}(2nm+n-3m^2-2m)-p_a^{m+1}(2n-4m)\\
& -p_a^m(2nm+n-3m^2+1)+p_a(n-m)+n-m+1.
\end{aligned}$$

The function $\Delta(m)$ has a root for $p_a=1$.

We first focus our attention on the case $m=2$ and study the condition $Var(Z_{a^2})-Var(Z_{a^3})>0$. We get

$$\begin{aligned}
\Delta(2) &= \frac{Var(Z_{a^2})-Var(Z_{a^3})}{p_a^2}\\
&= p_a^4(5n-16)-p_a^3(2n-8)-p_a^2(5n-11)+p_a(n-2)+n-1\\
&= (p_a-1)\left(p_a^3(5n-16)+p_a^2(3n-8)-p_a(2n+3)-n+1\right).
\end{aligned}$$

The four roots of this function have been computed with MAPLE: two roots are negative, one is $p_a = 1$, and one is positive $p_a = p^*$, where p^* is defined below. The closed form of p^* is too long to be reported here. We observe that function $\Delta(2)$ is positive in the interval $(0, p^*)$, which shrinks as n grows. For $n \to \infty$, $p^* = 0.6056592526$.

Repeating the analysis for $m = 3$, we obtain

$$\begin{aligned}\Delta(3) &= \frac{Var(Z_{a^3}) - Var(Z_{a^4})}{p_a^3} \\ &= p_a^5(7n-33) - p_a^4(2n-12) - p_a^3(7n-26) + n - 2 \\ &= (p_a - 1)\left(p_a^4(7n-33) + p_a^3(5n-21) - p_a^2(2n-5) - p_a(2n-5) - n + 2\right).\end{aligned}$$

It turns out that the interval for p_a in which $\Delta(3) > 0$ is larger than $(0, p^*)$. In fact, as m increases, the difference $Var(Z_{a^m}) - Var(Z_{a^{m+1}})$ becomes positive for larger and larger values of p_a.

Finally, when $m = (n-1)/2$, we get

$$\Delta\left(\frac{n-1}{2}\right) = \frac{n+3}{2} + \frac{p_a}{4}\left(p_a^{\frac{n+1}{2}}(n+1)^2 - 8p_a^{\frac{n-1}{2}} - p_a^{\frac{n-3}{2}}(1+6n+n^2) + 2n + 2\right),$$

and we can choose any p_a in the interval $(0, 1)$. To summarize, $p < 0.6$ assures the monotonicity for all n and $2 \le m \le (n+1)/2$. ■

Fact 3.12. *For any word y and for any $d \in \mathcal{P}(y)$,*

$$\prod_{j=m-d+1}^{m} p_{y_{[j]}} = \prod_{j=1}^{d} p_{y_{[j]}}.$$

Proof. Let us decompose $y = (uv)^k u$ where $|u| = d$. Then, clearly, y starts with uv and ends with vu, which have the same product of probabilities under the Bernoulli model. ■

The next three propositions are concerned with the monotonicity of the variance and the corresponding scores.

Fact 3.13. *Let w be a nonempty substring of a text generated by a Bernoulli process and wb a unit extension of w, $b \in \Sigma$. If $p_{max} < 1/\sqrt[m]{4m+2}$, then $Var(Z_{wb}) < Var(Z_w)$.*

Proof. Let $Z_i(w)$ be the indicator random variable that w occurs in the text x at position i. Then

$$Z_w = \sum_{i=1}^{n-m+1} Z_i(w), \qquad Z_{wb} = \sum_{i=1}^{n-m} Z_i(w) Z_{i+m}(b).$$

The proof is divided in two parts. The first is to show that $Var(Z_w) > Var\left(\sum_{i=1}^{n-m} Z_i(w)\right)$ when $p_{max} < 1/\sqrt[m]{2m-1}$. Then we prove that $Var\left(\sum_{i=1}^{n-m} Z_i(w)\right) > Var(Z_{wb})$ when $p_{max} < 1/\sqrt[m]{4m+2}$. Since $1/\sqrt[m]{4m+2} < 1/\sqrt[m]{2m-1}$, the conclusion holds when $p_{max} < 1/\sqrt[m]{4m+2}$.

Let us start with the first part. We have

$$Var(Z_w) = Var\left(\sum_{i=1}^{n-m} Z_i(w)\right) + \hat{p}(1-\hat{p}) + 2\sum_{i=1}^{n-m} Cov\left(Z_i(w), Z_{n-m+1}(w)\right).$$

Due to the independence

$$\begin{aligned}\sum_{i=1}^{n-m} Cov\left(Z_i(w), Z_{n-m+1}(w)\right) &= \sum_{i=n-2m+2}^{n-m} Cov\left(Z_i(w), Z_{n-m+1}(w)\right) \\ &\ge -(m-1)\hat{p}^2.\end{aligned}$$

Then

$$Var(Z_w) - Var\left(\sum_{i=1}^{n-m} Z_i(w)\right) \geq \hat{p}(1-\hat{p}) - 2(m-1)\hat{p}^2$$
$$= \hat{p}(1-(2m-1)\hat{p}).$$

Since $\hat{p} \leq p_{max}^m < 1/(2m-1)$, the first part of the proof follows.

Let us prove the second part. We have

$$Var\left(\sum_{i=1}^{n-m} Z_i(w)\right) - Var(Z_{wb}) = E\left(\left(\sum_{i=1}^{n-m}(Z_i(w)-\hat{p}) - \sum_{i=1}^{n-m}(Z_i(w)Z_{i+m}(b)-\hat{p}\,p_b)\right)\right.$$
$$\left.\cdot\left(\sum_{i=1}^{n-m}(Z_i(w)-\hat{p}) + \sum_{i=1}^{n-m}(Z_i(w)Z_{i+m}(b)-\hat{p}\,p_b)\right)\right)$$
$$= \sum_{i=1}^{n-m} Cov\left(Z_i(w)(1-Z_{i+m}(b)), Z_i(w)(1+Z_{i+m}(b))\right)$$
$$+ \sum_{i=1}^{n-m}\sum_{j\neq i} Cov\left(Z_i(w)(1-Z_{i+m}(b)), Z_j(w)(1+Z_{j+m}(b))\right)$$
$$= (n-m)(\hat{p}(1-p_b) - \hat{p}^2(1-p_b^2))$$
$$+ 2\sum_{i=1}^{n-m}\sum_{j=i+1}^{i+m}\left(E(Z_i(w)(1-Z_{i+m}(b))Z_j(w)(1+Z_{j+m}(b)) - \hat{p}^2(1-p_b^2)\right)$$
$$\geq (n-m)\hat{p}(1-p_b)(1-\hat{p}(1+p_b)) - 2(n-m)m\hat{p}^2(1-p_b^2)$$
$$= (n-m)\hat{p}(1-p_b)(1-\hat{p}(1+p_b) - 2m\hat{p}(1+p_b))$$
$$= (n-m)\hat{p}(1-p_b)(1-(2m+1)\hat{p}(1+p_b)).$$

Since $\hat{p} \leq p_{max}^m < 1/(4m+2)$, the second part follows, and also the conclusion. ■

Fact 3.14. *Let w be a nonempty substring of a text generated by a Bernoulli process, and wb a right extension of w, $b \in \Sigma$. If $p_{max} < \sqrt{2}-1$, then $\frac{E(Z_{wb})}{\sqrt{Var(Z_{wb})}} < \frac{E(Z_w)}{\sqrt{Var(Z_w)}}$.*

Proof. We define $\Delta(w,b) \equiv Var(Z_w)E(Z_{wb})^2 - Var(Z_{wb})E(Z_w)^2$. We have to prove $\Delta(w,b) < 0$. We have

$$\frac{\Delta(w,b)}{\hat{p}^2} = Var(Z_w)p_b^2(n-m)^2 - Var(Z_{wb})(n-m+1)^2$$
$$= (n-m)^2(p_b^2 Var(Z_w) - Var(Z_{wb})) - (2n-2m+1)Var(Z_{wb}).$$

First we evaluate $Var(Z_w)$, and we set $N = n-m$ for convenience.

$$Var(Z_w) = \hat{p}\left((N+1)(1-\hat{p}) - 2(m-1)\hat{p}(N+1-m/2) + 2B(w)\right)$$
$$\leq \hat{p}(N+1)\left(1-\hat{p}-2(m-1)\hat{p} + \frac{m(m-1)\hat{p}}{N+1} + \frac{2}{N+1}\sum_{l=1}^{m-1}(N+1-l)p_b^l\right)$$
$$= \hat{p}(N+1)\left(1-\hat{p}\left(2m-1+\frac{m(m-1)}{N+1}\right) + 2\sum_{l=1}^{m-1}\left(1-\frac{l}{N+1}\right)p_b^l\right)$$

implies that

$$\left(\frac{N}{N+1}\right)^2 \frac{p_b^2 Var(Z_w)}{\hat{p}\,p_b} \le p_b N \left(1 - \hat{p}\left(2m - 1 + \frac{m(m-1)}{N+1}\right) + 2\sum_{l=1}^{m-1}\left(1 - \frac{l}{N+1}\right) p_b^l\right).$$

Next we evaluate $Var(Z_{wb})$:

$$\begin{aligned}\frac{Var(Z_{wb})}{\hat{p}\,p_b} &= \left(N(1-\hat{p}\,p_b) - 2\hat{p}\,p_b\left(N - \frac{m+1}{2}\right)m + 2B(wb)\right)\\ &\ge N\left(1 - \hat{p}\,p_b - 2\hat{p}\,p_b\left(1 - \frac{m+1}{2N}\right)m\right).\end{aligned}$$

Note that since we are interested in the worst case for the difference $Var(Z_w) - Var(Z_{wb})$, we set $B(wb) = 0$ and $B(w)$ maximal. This happens when w is a word of the form a^m where a is the symbol with the highest probability p_{max} and $c \neq a$. Recall that Fact 3.10 says that $0 \le B(w) \le B(a^m)$. Then

$$\begin{aligned}\frac{\Delta(w,b)}{\hat{p}\,p_b(N+1)^2} &= \frac{\left(\frac{N}{N+1}\right)^2 p_b^2 Var(Z_w) - Var(Z_{wb})}{\hat{p}\,p_b}\\ &\le N\left(p_b - \hat{p}\,p_b\left(2m - 1 - \frac{m(m-1)}{N+1}\right) + 2p_b\sum_{l=1}^{m-1}\left(1 - \frac{l}{N+1}\right)p_b^l\right.\\ &\qquad \left. - 1 + \hat{p}\,p_b + 2\hat{p}\,p_b\left(1 - \frac{m+1}{2N}\right)m\right)\\ &= N\left(p_b - 1 + \hat{p}\,p_b\left(\frac{m(m-1)}{N+1} - \frac{m(m+1)}{N} + 2\right) + 2p_b\sum_{l=1}^{m-1}\left(1 - \frac{l}{N+1}\right)p_b^l\right)\\ &= N\left(p_b - 1 + \hat{p}\,p_b\left(2 - m\left(\frac{m+1}{N(N+1)} + \frac{2}{N+1}\right)\right) + 2p_b\sum_{l=1}^{m-1}\left(1 - \frac{l}{N+1}\right)p_b^l\right)\\ &\le N\left(p_b - 1 + 2\hat{p}\,p_b + 2p_b\sum_{l=1}^{m-1} p_b^l\right)\\ &\le N\left(p_{max} - 1 + 2p_{max}^{m+1} + 2p_{max}\sum_{l=1}^{m-1} p_{max}^l\right)\\ &= N\left(p_{max} - 1 + 2p_{max}\sum_{l=1}^{m} p_{max}^l\right)\\ &= N\left(-(p_{max}+1) + 2p_{max}\sum_{l=0}^{m} p_{max}^l\right)\\ &= N(1+p_{max})\left(-1 + 2p_{max}\frac{1 - p_{max}^{m+1}}{1 - p_{max}^2}\right).\end{aligned}$$

We used the fact that $p_b \le p_{max}$, $\hat{p} \le p_{max}^m$ and that $\frac{m+1}{N(N+1)} + \frac{2}{N+1} > 0$. A sufficient condition for the function $\Delta(w,b)$ to be negative is

$$2(1 - p_{max}^{m+1})p_{max} \le 1 - p_{max}^2.$$

TABLE 2. THE VALUE OF p^* FOR SEVERAL CHOICES OF m, FOR WHICH FUNCTION $\Delta(w, b)$ IS NEGATIVE IN THE INTERVAL $p_{max} \in (0, p^*)$. p^* CONVERGES TO $\sqrt{2} - 1$

m	p^*	m	p^*	m	p^*
2	0.4406197005	5	0.4157303841	30	0.4142135624
3	0.4238537991	10	0.4142316092	50	0.4142135624
4	0.4179791697	20	0.4142135651	100	0.4142135624

Table 2 shows the root p^* of $2(1 - p_{max}^{m+1})p_{max} - 1 + p_{max}^2 = 0$ when $p_{max} \in [0, 1]$. For large m, it suffices to show that $2p_{max} \leq 1 - p_{max}^2$, which corresponds to $p_{max} \leq \sqrt{2} - 1$. ■

Theorem 3.3. *Let x be a text generated by a Bernoulli process. If $f(w) = f(wv)$ and $p_{max} < \min\{1/\sqrt[m]{4m}, \sqrt{2} - 1\}$, then*

$$\frac{f(wv) - E(Z_{wv})}{\sqrt{Var(Z_{wv})}} > \frac{f(w) - E(Z_w)}{\sqrt{Var(Z_w)}}.$$

Proof. The choice $N(w) = \sqrt{Var(Z_w)}$, $\rho(w) = E(w)/\sqrt{Var(Z_w)}$ satisfies the conditions of Theorem 3.1 because the bound on p_{max} satisfies the hypothesis of Facts 3.13 and 3.14. ■

An interesting observation by Sinha and Tompa (2000) is that the score in Theorem 3.3 obeys the following relation:

$$z(w) \leq \frac{f(w) - E(Z_w)}{\sqrt{E(Z_w) - E(Z_w)^2}} \quad \text{when} \quad E(Z_w) - E(Z_w)^2 > 0$$

since $Var(Z_w) \geq E(Z_w) - E(Z_w)^2$ (see Sinha and Tompa [2000] for details). It is therefore sufficient to know $E(Z_w)$ to have an upper bound of the score. If the bound happens to be smaller than than the threshold, then the algorithm can disregard that word, avoiding the computation of the exact variance.

Theorem 3.4. *Let x be a text generated by a Bernoulli process.*
If $f(w) = f(wv) \equiv f$ and $p_{max} < \min\{1/\sqrt[m]{4m}, \sqrt{2} - 1\}$, then

$$\left|\frac{f(wv) - E(Z_{wv})}{\sqrt{Var(Z_{wv})}}\right| < \left|\frac{f(w) - E(Z_w)}{\sqrt{Var(Z_w)}}\right| \quad \textit{iff} \quad f > E(Z_w)\frac{\gamma\sqrt{Var(Z_w)} + \sqrt{Var(Z_{wv})}}{\sqrt{Var(Z_w)} + \sqrt{Var(Z_{wv})}}$$

where $\gamma = E(Z_{wv})/E(Z_w)$.

Proof. The choice $N(w) = \sqrt{Var(Z_w)}$, $\rho(w) = E(w)/\sqrt{Var(Z_w)}$ satisfies the conditions of Theorem 3.2 because the bound on p_{max} satisfies the hypothesis of Facts 3.13 and 3.14. ■

Table 3 collects these properties.

3.2. The expected number of occurrences under Markov models

Fact 3.15. *Let w and v be two nonempty substrings of a text generated by a Markov process of order $M > 0$. Then $\hat{E}(Z_{wv}) \leq \hat{E}(Z_w)$.*

Proof. Let us first prove the case $M = 1$ for simplicity. Recall that an estimator of the expected count when $M = 1$ is given by

$$\hat{E}(Z_w) = \frac{f(w_{[1,2]})f(w_{[2,3]})\ldots f(w_{[|w|-1,|w|]})}{f(w_{[2]})f(w_{[3]})\ldots f(w_{[|w|-1]})}.$$

TABLE 3. 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*
(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)	$\dfrac{f(wv)}{E(Z_{wv})} > \dfrac{f(w)}{E(Z_w)}$	$f(w) = f(wv)$
(2.4)	$\dfrac{f(wv) - E(Z_{wv})}{E(Z_{wv})} > \dfrac{f(w) - E(Z_w)}{E(Z_w)}$	$f(w) = f(wv)$
(2.5)	$\dfrac{f(wv) - E(Z_{wv})}{\sqrt{E(Z_{wv})}} > \dfrac{f(w) - E(Z_w)}{\sqrt{E(Z_w)}}$	$f(w) = f(wv)$
(2.6)	$\left\lvert\dfrac{f(wv) - E(Z_{wv})}{\sqrt{E(Z_{wv})}}\right\rvert > \left\lvert\dfrac{f(w) - E(Z_w)}{\sqrt{E(Z_w)}}\right\rvert$	$f(w) = f(wv), \quad f(w) > E(Z_w)\sqrt{\gamma}$
(2.7)	$\dfrac{(f(wv) - E(Z_{wv}))^2}{E(Z_{wv})} > \dfrac{(f(w) - E(Z_w))^2}{E(Z_w)}$	$f(w) = f(wv), \quad f(w) > E(Z_w)\sqrt{\gamma}$
(2.8)	$\dfrac{f(wv) - E(Z_{wv})}{\sqrt{E(Z_{wv})(1 - \hat{p}\hat{q})}} > \dfrac{f(w) - E(Z_w)}{\sqrt{E(Z_w)(1 - \hat{p})}}$	$f(w) = f(wv), \quad \hat{p} < 1/2$
(2.9)	$Var(Z_{wv}) < Var(Z_w)$	$p_{max} < 1/\sqrt[m]{4m}$
(2.10)	$\dfrac{E(Z_{wv})}{\sqrt{Var(Z_{wv})}} < \dfrac{E(Z_w)}{\sqrt{Var(Z_w)}}$	$p_{max} < \sqrt{2} - 1$
(2.11)	$\dfrac{f(wv) - E(Z_{wv})}{\sqrt{Var(Z_{wv})}} > \dfrac{f(w) - E(Z_w)}{\sqrt{Var(Z_w)}}$	$f(w) = f(wv), \quad p_{max} < \min\{1/\sqrt[m]{4m}, \sqrt{2} - 1\}$
(2.12)	$\left\lvert\dfrac{f(wv) - E(Z_{wv})}{\sqrt{Var(Z_{wv})}}\right\rvert > \left\lvert\dfrac{f(w) - E(Z_w)}{\sqrt{Var(Z_w)}}\right\rvert$	$f(w) = f(wv), \quad p_{max} < \min\{1/\sqrt[m]{4m}, \sqrt{2} - 1\}$ and $f(w) > E(Z_w)\dfrac{\gamma\sqrt{Var(Z_w)} + \sqrt{Var(Z_{wv})}}{\sqrt{Var(Z_w)} + \sqrt{Var(Z_{wv})}}$

Let us evaluate

$$\frac{\hat{E}(Z_{wv})}{\hat{E}(Z_w)} = \frac{\dfrac{f(w_{[1,2]})f(w_{[2,3]})\dots f(w_{[|w|-1,|w|]})f(w_{[|w|]}v_{[1]})f(v_{[1,2]})\dots f(v_{[|v|-1,|v|]})}{f(w_{[2]})f(w_{[3]})\dots f(w_{[|w|-1]})f(w_{[|w|]})f(v_{[1]})\dots f(v_{[|v|-1]})}}{\dfrac{f(w_{[1,2]})f(w_{[2,3]})\dots f(w_{[|w|-1,|w|]})}{f(w_{[2]})f(w_{[3]})\dots f(w_{[|w|-1]})}}$$

$$= \frac{f(w_{[|w|]}v_{[1]})f(v_{[1,2]})\dots f(v_{[|v|-1,|v|]})}{f(w_{[|w|]})f(v_{[1]})\dots f(v_{[|v|-1]})}.$$

Note that numerator and denominator have the same number of factors and that $f(w_{[|w|]}v_{[1]}) \le f(w_{[|w|]})$, $f(v_{[1,2]}) \le f(v_{[1]})$, $\dots$, $f(v_{[|v|-1,|v|]}) \le f(v_{[|v|-1]})$. Therefore,

$$\frac{\hat{E}(Z_{wv})}{\hat{E}(Z_w)} \le 1.$$

Suppose now we have a Markov chain of order $M > 1$. Using a standard procedure, we can transform it into a Markov model of order one. The alphabet of the latter is composed of symbols in one-to-one correspondence with all the possible substrings of length $M - 1$.

Since the argument above is independent from the size of the alphabet, the conclusion holds for any Markov chain. ■

Fact 3.16. *Let x be text generated by a Markov process of order $M > 0$. If $f(w) = f(wv)$, then*

1. $f(wv) - \hat{E}(Z_{wv}) \geq f(w) - \hat{E}(Z_w)$,
2. $\dfrac{f(wv)}{\hat{E}(Z_{wv})} \geq \dfrac{f(w)}{\hat{E}(Z_w)}$,
3. $\dfrac{f(wv) - \hat{E}(Z_{wv})}{\hat{E}(Z_{wv})} \geq \dfrac{f(w) - \hat{E}(Z_w)}{\hat{E}(Z_w)}$,
4. $\dfrac{f(wv) - \hat{E}(Z_{wv})}{\sqrt{\hat{E}(Z_{wv})}} \geq \dfrac{f(w) - \hat{E}(Z_w)}{\sqrt{\hat{E}(Z_w)}}$.

Proof. Directly from Theorem 3.1 and Fact 3.15. ■

Fact 3.17. *Let x be text generated by a Markov process of order $M > 0$. If $f(w) = f(wv) \equiv f$, then*

1. $\left|\dfrac{f(wv) - \hat{E}(Z_{wv})}{\sqrt{\hat{E}(Z_{wv})}}\right| \geq \left|\dfrac{f(w) - \hat{E}(Z_w)}{\sqrt{\hat{E}(Z_w)}}\right|$ *iff* $f > E(Z_w)\sqrt{\gamma}$,
2. $\dfrac{(f(wv) - \hat{E}(Z_{wv}))^2}{\hat{E}(Z_{wv})} \geq \dfrac{(f(w) - \hat{E}(Z_w))^2}{\hat{E}(Z_w)}$ *iff* $f > E(Z_w)\sqrt{\gamma}$

where $\gamma = E(Z_{wv})/E(Z_w)$.

Proof. Directly from Fact 3.3 and Fact 3.15. ■

3.3. The expected number of colors for Bernoulli and Markov models

Fact 3.18. *Let w and v be two nonempty substrings of a text generated by a any process. Then $E(W_{wv}) \leq E(W_w)$.*

Proof. Recall that

$$E(W_w) = k - \sum_{j=1}^{k} \mathbf{P}[Z_w^j = 0]$$

where Z_w^j represents the number of occurrences of the word w in th j-th sequence. Since we have

$$\mathbf{P}[Z_{wv}^j = 0] = \mathbf{P}[Z_w^j = 0] + \mathbf{P}[Z_w^j \neq 0 \text{ and } Z_{wv}^j = 0],$$

then

$$E(W_w) - E(W_{wv}) = \sum_{j=1}^{k} \mathbf{P}[Z_w^j \neq 0 \text{ and } Z_{wv}^j = 0] \geq 0$$

and therefore the conclusion follows. ■

The following three facts are a direct consequence of Fact 3.1 and Fact 3.18.

Fact 3.19. *Let x be a text generated by any process. If $c(w) = c(wv)$, then*

1. $c(wv) - E(W_{wv}) \geq c(w) - E(W_w)$,
2. $\dfrac{c(wv)}{E(W_{wv})} \geq \dfrac{c(w)}{E(W_w)}$,
3. $\dfrac{c(wv) - E(W_{wv})}{E(W_{wv})} \geq \dfrac{c(w) - E(W_w)}{E(W_w)}$,
4. $\dfrac{c(wv) - E(W_{wv})}{E(W_{wv})} \geq \dfrac{c(w) - E(W_w)}{E(W_w)}$.

Proof. Directly from Theorem 3.1 and Fact 3.18. ■

Fact 3.20. *Let x be a text generated by any process. If $c(w) = c(wv) \equiv c$, then*

1. $\left|\dfrac{c(wv) - E(W_{wv})}{\sqrt{E(W_{wv})}}\right| \geq \left|\dfrac{c(w) - E(W_w)}{\sqrt{E(W_w)}}\right|$ *iff* $c > E(W_w)\sqrt{\gamma}$,
2. $\dfrac{(c(wv) - E(W_{wv}))^2}{E(W_{wv})} \geq \dfrac{(c(w) - E(W_w))^2}{E(W_w)}$ *iff* $c > E(W_w)\sqrt{\gamma}$

where $\gamma = E(W_{wv})/E(W_w)$.

Proof. Directly from Fact 3.3 and Fact 3.18. ■

Tables 4 and 5 summarize the collection of these properties.

4. COMPUTING EQUIVALENCE CLASSES AND SCORES

Here we pursue substring partitions $\{C_1, C_2, \ldots, C_l\}$ in forms which would enable us to restrict the computation of the scores to a constant number of candidates in each class C_i. Specifically, 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; and all w in C_i to 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 begin by recalling a few basic facts and constructs from, e.g., Blumer *et al.* (1987). The experienced reader may skip most of this part. 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 words. Also, by definition, if $y \equiv_l w$, then $f(y) = f(w)$.

For instance, in the string `ataatataataatataatatag` the set {`ataa`, `ataat`, `ataata`} is a left-equivalent class (with position set {1, 6, 9, 14}) and so are {`taa`, `taat`, `taata`} and {`aa`, `aat`, `aata`}. We have 39 left-equivalent classes, much less than the total number of substrings, which is $22 \times 23/2 = 253$, and than the number of distinct substrings, in this case 61.

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 (Blumer *et al.*, 1987; Clift *et al.*, 1986). 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 the following.

TABLE 4. 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*
(3.1)	$\hat{E}(Z_{wv}) \le \hat{E}(Z_w)$	none
(3.2)	$f(wv) - \hat{E}(Z_{wv}) \ge f(w) - \hat{E}(Z_w)$	$f(w) = f(wv)$
(3.3)	$\dfrac{f(wv)}{\hat{E}(Z_{wv})} \ge \dfrac{f(w)}{\hat{E}(Z_w)}$	$f(w) = f(wv)$
(3.4)	$\dfrac{f(wv) - \hat{E}(Z_{wv})}{\hat{E}(Z_{wv})} \ge \dfrac{f(w) - \hat{E}(Z_w)}{\hat{E}(Z_w)}$	$f(w) = f(wv)$
(3.5)	$\dfrac{f(wv) - \hat{E}(Z_{wv})}{\sqrt{\hat{E}(Z_{wv})}} \ge \dfrac{f(w) - \hat{E}(Z_w)}{\sqrt{\hat{E}(Z_w)}}$	$f(w) = f(wv)$
(3.6)	$\left\lvert \dfrac{f(wv) - \hat{E}(Z_{wv})}{\sqrt{\hat{E}(Z_{wv})}} \right\rvert \ge \left\lvert \dfrac{f(w) - \hat{E}(Z_w)}{\sqrt{\hat{E}(Z_w)}} \right\rvert$	$f(w) = f(wv), \quad f(w) > E(Z_w)\sqrt{\gamma}$
(3.7)	$\dfrac{(f(wv) - \hat{E}(Z_{wv}))^2}{\hat{E}(Z_{wv})} \ge \dfrac{(f(w) - \hat{E}(Z_w))^2}{\hat{E}(Z_w)}$	$f(w) = f(wv), \quad f(w) > E(Z_w)\sqrt{\gamma}$

TABLE 5. MONOTONICITIES OF THE SCORES ASSOCIATED WITH THE NUMBER OF COLORS c UNDER ANY MODEL FOR THE RANDOM VARIABLE W; WE SET $\gamma \equiv E(W_{wv})/E(W_w)$

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

Lemma 4.1. *The equivalence relation $\equiv_x$ is the transitive closure of $\equiv_l \cup \equiv_r$.*

More importantly, 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 $n + 1$.

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, for which we refer to, e.g., Apostolico *et al.* (2000), Apostolico and Galil (1997), and Blumer *et al.* (1987) with their quoted literature, or easy adaptations thereof. The graph for $\equiv_l$, for instance, is the compact subword tree T_x of x, whereas the graph for $\equiv_r$ is the *dawg*, or *directed acyclic word graph* D_x, for x. The graph for $\equiv_x$ is the compact version of the the *dawg*.

These data structures are known to commute in simple ways, so that, say, an $\equiv_x$-class can be found on T_x as the union of some left-equivalent classes or, alternatively, as the union of some right-equivalent classes. Following are some highlights for the inexperienced reader. Beginning with left-equivalent classes that correspond one-to-one to the nodes of T_x, we can build some right-equivalent classes as follows. We use the elementary fact that whenever there is a branching node μ in T_x corresponding to $w = ay, a \in \Sigma$, then there is also a node ν corresponding to y, and there is a special *suffix* link directed from ν to μ. Such auxiliary links induce another tree on the nodes of T_x that we may call S_x. It is now easy to find a right-equivalent class with the help of suffix links. For this, we traverse S_x bottom-up while grouping in a single class all strings such that their terminal nodes in T_x are roots of isomorphic subtrees of T_x. When a subtree that violates the isomorphism condition is encountered, we are at the end of one class and we start with a new one.

For example, the three subtrees rooted at the solid nodes in Fig. 2 correspond to the end-sets of `ataata`, `taata` and `aata`, which are the same, namely, {6, 11, 14, 19}. These three words define the right-equivalent class {`ataata`, `taata`, `aata`}. In fact, this class cannot be made larger because the two

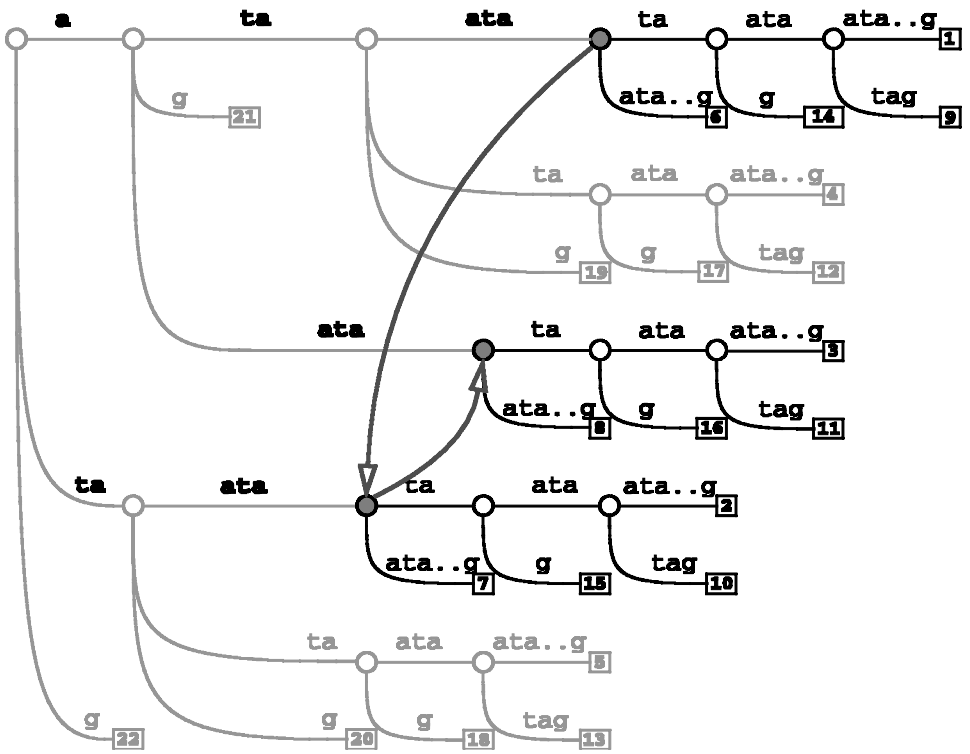

FIG. 2. The tree T_x for $x =$ `ataatataataatataatatag`: subtrees rooted at the solid nodes are isomorphic.

subtrees rooted at the end nodes of ata and tataata are not isomorphic to the substree of the class. We leave it as an exercise for the reader to find *all* the right-equivalence classes on T_x. It turns out that there are 24 such classes in this example.

Subtree isomorphism is checked by a classical linear-time algorithm by Aho *et al.* (1974). But on T_x this is done even more quickly once the f counts are available (Apostolico and Lonardi, 2002; Gusfield, 1997).

Lemma 4.2. *Let T_1 and T_2 be two subtrees of T_x. T_1 and T_2 are isomorphic if and only if they have the same number of leaves and their roots are connected by a chain of suffix links.*

Proof. If T_1 and T_2 are isomorphic, then clearly they have same number of leaves. Also, if they were not linked by a chain of suffix links, strings w_1 and w_2 corresponding to the path-labels of the roots of T_1 and T_2 could not be a suffix of one another. Hence, their end-sets would be different, contrary to the hypothesis of the isomorphism of the subtrees T_1 and T_2.

Let us assume, w.l.o.g., that there is a chain formed by l suffix links from the root of T_1 to the root of T_2, $l \geq 1$. Let uw be the path-label for the root of T_1, and w the path-label for the root of T_2, whence $l = |u|$. In general, we have that $endpos(uw) \subseteq endpos(w)$. Since we know that $f(uw) = f(w)$, then the only possibility is that $endpos(uw) = endpos(w)$; hence, the subtrees are isomorphic. ■

If, during the bottom-up traversal of S_x, we put in the same class strings such that their terminal *arc* leads to nodes with the same frequency counts f, then this would identify and produce the $\equiv_x$-classes, i.e., the smallest substring partition.

For instance, starting from the right-equivalent class $C = \{$ataata, taata, aata$\}$, one can augment it with of all words which are left-equivalent to the elements of C. The result is one $\equiv_x$-class composed by {ataa, ataat, ataata, taa, taat, taata, aa, aat, aata}. Their respective *pos* sets are {1, 6, 9, 14}, {1, 6, 9, 14}, {1, 6, 9, 14}, {2, 7, 10, 15}, {2, 7, 10, 15}, {2, 7, 10, 15}, {3, 8, 11, 16}, {3, 8, 11, 16}, {3, 8, 11, 16}. Their respective *endpos* sets are {4, 9, 12, 17}, {5, 10, 13, 18}, {6, 11, 14, 19}, {4, 9, 12, 17}, {5, 10, 13, 18}, {6, 11, 14, 19}, {4, 9, 12, 17}, {5, 10, 13, 18}, {6, 11, 14, 19}. Because of Lemma 4.1, given two words y and w in the class, either they share the start set, or they share the end set, or they share the start set by transitivity with a third word in the class, or they share the end set by transitivity with a third word in the class. It turns out that there are only seven $\equiv_x$-classes in our example.

Note that the longest string in this $\equiv_x$-class is unique (ataata) and that it contains all the others as substrings. The shortest string is unique as well (aa). As said, the number of occurrences for all the words in the same class is the same (four in the example). Figure 3 illustrates the seven equivalence classes for our running example. The words in each class have been organized in a lattice, where edges correspond to extensions (or contractions) of a single symbol. In particular, horizontal edges correspond to right extensions and vertical edges to left extensions.

While the longest word in an $\equiv_x$-class is unique, there may be in general more than one shortest word. Consider for example the text $x = \mathtt{a}^k\mathtt{g}^k$, with $k > 0$ (see Fig. 4). 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.

We recall that by Lemma 4.1 each $\equiv_x$-class C can be expressed as the union of one or more left-equivalent classes. Alternatively, C can be also expressed as the union of one or more right-equivalent classes. The example above shows that there are cases in which we *cannot* merge left- or right-equivalent classes without violating the uniqueness of the shortest word. Thus, we may use the $\equiv_x$-classes as the C_i's in our partition only if we are interested in detecting overrepresented words. If underrepresented words are also wanted, then we must represent the same $\equiv_x$-class once for each distinct shortest word in it.

It is not difficult to accommodate this in our subtree merge procedure. Let $p(u)$ denote the parent of u in T_x. While traversing S_x bottom-up, we merge two nodes u and v with the same f count if and only if u and v are connected by a suffix link and $p(u)$ and $p(v)$ are also. This results in a substring partition slightly coarser $\equiv_x$, which will be denoted by $\tilde{\equiv}_x$. In conclusion, we can state the following fact.

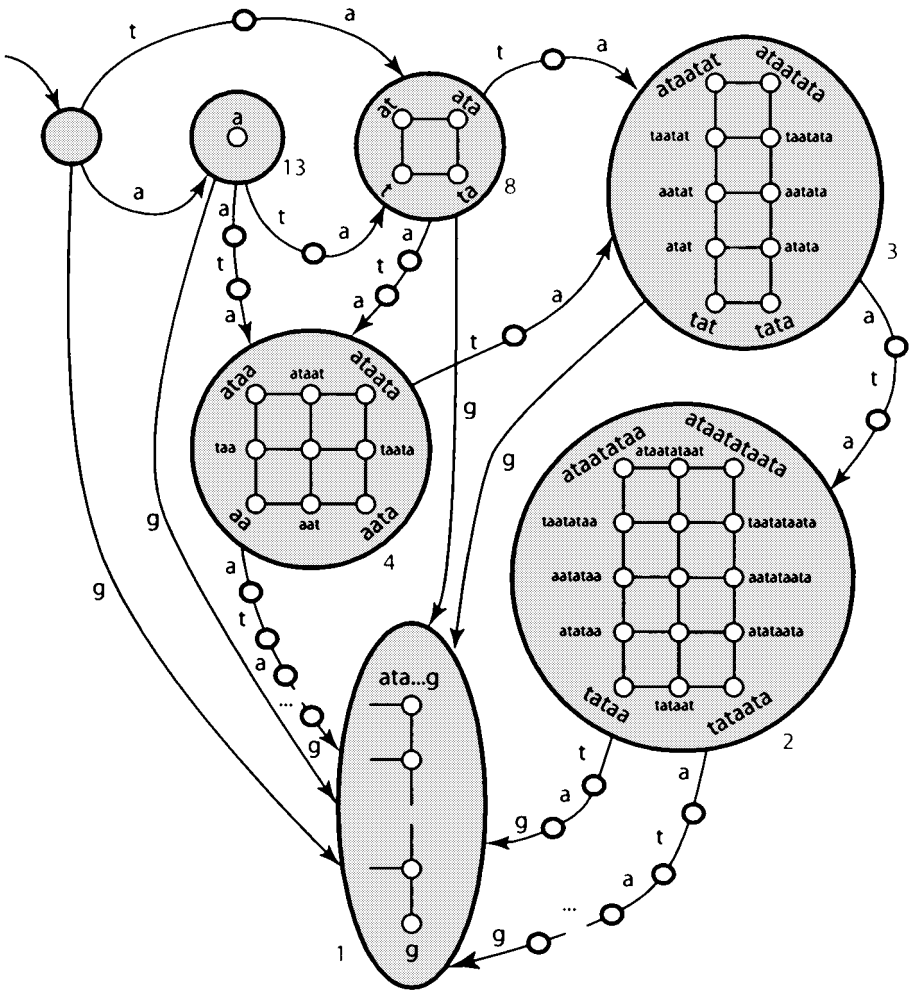

FIG. 3. A representation of the seven $\equiv_x$-classes for $x = \texttt{ataatataataatataatatag}$. The words in each class can be organized in a lattice. Numbers refer to the number of occurrences.

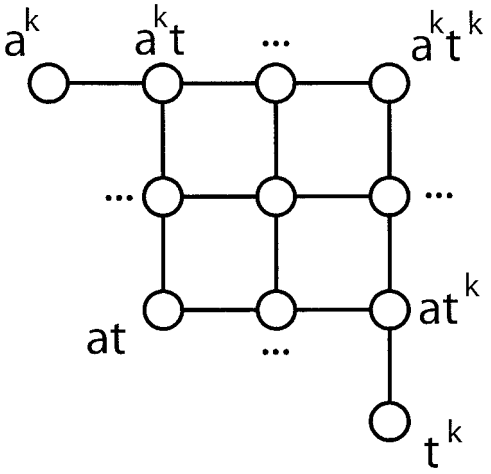

FIG. 4. One $\equiv_x$-class for the string $x = \texttt{a}^k\texttt{t}^k$.

Fact 4.1. *Let* $\{C_1, C_2, \ldots, C_l\}$ *be the set of equivalence classes built on the equivalence relation* $\tilde{\equiv}_x$ *on the substrings of text* x. *Then, for all* $1 \leq i \leq l$,

1. $\max(C_i)$ *and* $\min(C_i)$ *are unique,*
2. *all* $w \in C_i$ *are on some* $(\min(C_i), \max(C_i))$*-path,*
3. *all* $w \in C_i$ *have the same number of occurrences* $f(w)$,
4. *all* $w \in C_i$ *have the same number of colors* $c(w)$.

We are now ready to address the computational complexity of our constructions. In Apostolico *et al.* (2000), linear-time algorithms are given to compute and store expected value $E(Z)$ and variance $Var(Z)$ for the number of occurrences under the Bernoulli model of *all* prefixes of a given string. The crux of that construction rests on deriving an expression of the variance (see Expression 1) that can be cast within the classical linear time computation of the "failure function" or smallest periods for all prefixes of a string (see, e.g., Aho *et al.* [1974]). These computations are easily adapted to be carried out on the linked structure of graphs such as S_x or D_x, thereby yielding expectation and variance values at all nodes of T_x, D_x, or the compact variant of the latter. These constructions take time and space linear in the size of the graphs, hence, linear in the length of x. Combined with our monotonicity results this yields immediately:

Theorem 4.1. *Under the Bernoulli models, the sets* $\mathcal{O}_z^T$ *and* $\mathcal{U}_z^T$ *for scores*

$$z_1(w) = f(w) - E(Z_w)$$

$$z_2(w) = \frac{f(w)}{E(Z_w)}$$

$$z_3(w) = \frac{f(w) - E(Z_w)}{E(Z_w)}$$

$$z_4(w) = \frac{f(w) - E(Z_w)}{\sqrt{E(Z_w)}}$$

$$z_5(w) = \frac{f(w) - E(Z_w)}{\sqrt{E(Z_w)(1 - \hat{p})}} \quad (when\ \hat{p} < 1/2)$$

$$z_6(w) = \frac{f(w) - E(Z_w)}{\sqrt{Var(Z_w)}} \quad (when\ p_{max} < \min\{1/\sqrt[m]{4m}, \sqrt{2} - 1\})$$

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

$$z_7(w) = \left| \frac{f(w) - E(Z_w)}{\sqrt{E(Z_w)}} \right|$$

$$z_8(w) = \frac{(f(w) - E(Z_w))^2}{E(Z_w)}$$

$$z_9(w) = \left| \frac{f(w) - E(Z_w)}{\sqrt{Var(Z_w)}} \right| \quad (when\ p_{max} < \min\{1/\sqrt[m]{4m}, \sqrt{2} - 1\})$$

can be computed in linear time and space.

The computation of $\hat{E}(Z_y)$ is more involved in Markov models than with Bernoulli. Recall from Expression 2 that the maximum likelihood estimator for the expectation is

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

where M is the order of the Markov chain. If we compute the (Markov) prefix product $pp(i)$ as

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

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) gives 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 auxiliary products $pp(i)$ in overall linear time, e.g., beginning at the node of T_x which is found at the end of the path to $x_{[1,M+1]}$ and then alternating between suffix- and direct edge transitions on the tree. We leave the details for an exercise. 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 these results.

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

$$z_{11}(w) = f(w) - \hat{E}(Z_w)$$

$$z_{12}(w) = \frac{f(w)}{\hat{E}(Z_w)}$$

$$z_{13}(w) = \frac{f(w) - \hat{E}(Z_w)}{\hat{E}(Z_w)}$$

$$z_{14}(w) = \frac{f(w) - \hat{E}(Z_w)}{\sqrt{\hat{E}(Z_w)}}$$

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

$$z_{15}(w) = \left| \frac{f(w) - E(Z_w)}{\sqrt{E(Z_w)}} \right|$$

$$z_{16}(w) = \frac{(f(w) - E(Z_w))^2}{E(Z_w)}$$

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 $\{E(Z_y^j) : j \in [1 \ldots k]\}$, 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 all $E(Z_y^j)$ are available, we can use Equation 3 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 $n = \sum_{i=1}^{k} |x^{(i)}|$. 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 any model, the sets $\mathcal{O}_z^T$ and $\mathcal{U}_z^T$ of a multisequence $\{x^{(1)}, x^{(2)}, \ldots, x^{(k)}\}$ for scores*

$$z_{17}(w) = c(w) - E(W_w)$$

$$z_{18}(w) = \frac{c(w)}{E(W_w)}$$

$$z_{19}(w) = \frac{c(w) - E(W_w)}{E(W_w)}$$

$$z_{20}(w) = \frac{c(w) - E(W_w)}{\sqrt{E(W_w)}}$$

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

$$z_{21}(w) = \left| \frac{c(w) - E(W_w)}{\sqrt{E(W_w)}} \right|$$

$$z_{22}(w) = \frac{(c(w) - E(W_w))^2}{E(W_w)}$$

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

5. CONCLUSIONS

We have shown that under several scores and models, we can bound the number of candidate over- and underrepresented 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 nondecreasing. 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 $\{C_1, C_2, \ldots, C_l\}$ that satisfy properties such as in Fact 4.1. Clearly, the equivalence relations $\equiv_l$, $\equiv_r$, and $\tilde{\equiv}_x$ 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 (respectively, 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 associated with the equivalence relation $\tilde{\equiv}_x$, which achieves the smallest number of classes under the constrains of Fact 4.1. 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.

ACKNOWLEDGMENTS

The passage by J.L. Borges which inspired the title of Apostolico (2001) was pointed out to the author by Gustavo Stolovitzky. We are also grateful to the referees for their helpful comments. In particular, we thank one of the referees for suggesting an alternative proof of Fact 3.13. Dan Gusfield brought to our attention that Lemma 4.2 had been previously established by Gusfield (1997).

REFERENCES

Aho, A.V., Hopcroft, J.E., and Ullman, J.D. 1974. *The Design and Analysis of Computer Algorithms*, Addison-Wesley, Reading, MA.

Apostolico, A. 2001. Of maps bigger than the empire. *Keynote, in Proc. 8th Int. Colloquium on String Processing and Information Retrieval* (Laguna de San Rafael, Chile, November 2001), IEEE Computer Society Press.

Apostolico, A., Bock, M.E., Lonardi, S., and Xu, X. 2000. Efficient detection of unusual words. *J. Comp. Biol.* 7(1–2), 71–94.

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

Apostolico, A., and Galil, Z., eds. 1997. *Pattern Matching Algorithms*, Oxford University Press, New York.

Apostolico, A., and Lonardi, S. 2001. Verbumculus. *www.cs.ucr.edu/~stelo/Verbumculus.*

Apostolico, A., and Lonardi, S. 2002. A speed-up for the commute between subword trees and DAWGs. *Information Processing Letters* 83(3), 159–161.

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

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

Clift, B., Haussler, D., McConnell, R., Schneider, T.D., and Stormo, G.D. 1986. Sequences landscapes. *Nucl. Acids Res.* 14, 141–158.

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

Gusfield, D. 1997. *Algorithms on Strings, Trees, and Sequences: Computer Science and Computational Biology*, Cambridge University Press, London.

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

Leung, M.Y., Marsh, G.M., and Speed, T.P. 1996. Over and underrepresentation of short DNA words in herpesvirus genomes. *J. Comp. Biol.* 3, 345–360.

Lonardi, S. 2001. *Global Detectors of Unusual Words: Design, Implementation, and Applications to Pattern Discovery in Biosequences.* Ph.D Thesis, Department of Computer Sciences, Purdue University.

Lundstrom, R. 1990. *Stochastic models and statistical methods for DNA sequence data.* Ph.D Thesis, University of Utah.

Pevzner, P.A., Borodovsky, M.Y., and Mironov, A.A. 1989. 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. Dyn.* 6, 1013–1026.

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

Reinert, G., Schbath, S., and Waterman, M.S. 2000. Probabilistic and statistical properties of words: An overview. *J. Comp. Biol.* 7, 1–46.

Sinha, S., and Tompa, M. 2000. A statistical method for finding transcription factor binding sites. *Proc. 8th Int. Conf. Intelligent Systems for Molecular Biology*, 344–354.

Stückle, E., Emmrich, C., Grob, U., and Nielsen, P. 1990. Statistical analysis of nucleotide sequences. *Nucl. Acids Res.* 18(22), 6641–6647.

Waterman, M.S. 1995. *Introduction to Computational Biology*, Chapman and Hall, London.

Address correspondence to:
Alberto Apostolico
Department of Computer Sciences
Purdue University
Computer Sciences Building
West Lafayette, IN 47907

E-mail: axa@cs.purdue.edu