Exact and Efficient Computation of the Expected Number of Missing and Common Words in Random Texts

Sven Rahmann[1] and Eric Rivals[2]

[1] Theoretische Bioinformatik (TBI)
Deutsches Krebsforschungszentrum (DKFZ),
Im Neuenheimer Feld 280, D-69120 Heidelberg, Germany,
`S.Rahmann@dkfz-heidelberg.de`

[2] L.I.R.M.M.
161 rue Ada, F-34392 Montpellier Cedex 5, France,
`rivals@lirmm.fr`

Abstract. The number of missing words (NMW) of length q in a text, and the number of common words (NCW) of two texts are useful text statistics. Knowing the distribution of the NMW in a random text is essential for the construction of so-called monkey tests for pseudorandom number generators. Knowledge of the distribution of the NCW of two independent random texts is useful for the average case analysis of a family of fast pattern matching algorithms, namely those which use a technique called q-gram filtration. Despite these important applications, we are not aware of any exact studies of these text statistics. We propose an efficient method to compute their expected values exactly. The difficulty of the computation lies in the strong dependence of successive words, as they overlap by $(q-1)$ characters. Our method is based on the enumeration of all string autocorrelations of length q, i.e., of the ways a word of length q can overlap itself. For this, we present the first efficient algorithm. Furthermore, by assuming the words are independent, we obtain very simple approximation formulas, which are shown to be surprisingly good when compared to the exact values.

1 Introduction

We consider *random texts*. A *text* of length n is a string of n characters from a given *alphabet* Σ of size σ. *Randomness* in this article refers to the symmetric Bernoulli model, meaning that the probability to encounter any character at any position of the text is $1/\sigma$, independently of the other text positions. As soon as the text length has been fixed to n, every text has the same chance of occurring, namely σ^{-n}. A *word* of length q, also called a *q-gram*, is a substring of length q. If $n \geq q$, we find exactly $(n-q+1)$ (overlapping) q-grams in a text of length n; they end at positions $q, q+1, \ldots, n$. However, not all of these need to be different.

We consider two applications that motivate why it is interesting to determine the distribution, and hence especially the expectation, of the number of missing words (NMW) in a random text, and of the number of common words (NCW) of two independent random texts.

Missing Words and Monkey Tests. Knowing the distribution of the NMW in a random text allows the construction of so-called *monkey tests* for pseudorandom number generators (PRNGs).

Assume we are given a subroutine that supposedly produces instances of a random variable U uniformly distributed in the interval $[0, 1]$, but we are unsure of the quality of this PRNG, i.e., whether the produced sequences of pseudorandom numbers indeed share properties with truly random sequences. In this case one usually runs a series of empirical tests, a list of which can be found in Knuth's comprehensive work [9].

One set of tests called *monkey tests* was proposed by Marsaglia and Zaman [10]. One variant works as follows: Each call to the PRNG is used to create a pseudorandom bit (e.g., the most or least significant bit of U), and a text of length n is created by concatenating the bits from successive calls to the PRNG. (Imagine a monkey typing randomly on a 0-1-keyboard; hence the name monkey test.) One counts how many of the 2^q possible q-grams are missing from the resulting text, and compares this number to the expected NMW in a truly random bit sequence. The PRNG should be rejected if these numbers differ significantly.

The advantage of monkey tests is that, for each q-gram, only one bit needs to be stored to indicate whether the q-gram has occurred in the text or not. Other tests, like the well-known chi-square test on the frequencies of q-grams, require the storage of an integer (the number of occurrences) for every q-gram. Hence monkey tests allow to choose q relatively large, such as $q = 33$, needing 2^{33} bits or 1 GB of memory. In comparison, if one runs the chi-square test with 1 GB of memory, one is restricted to $q \leq 28$ (assuming 4 bytes per integer). Hence the monkey test allows to detect possible deficiencies of the PRNG within a larger context and can capture dependencies within the generated sequence that other tests may miss.

To construct a precise statistical test, we need to know the distribution of the NMW. Here, we present an efficient method to compute the exact expectation. In [10] the authors express a conjecture about the distribution from simulations. In Section 4, we propose a slightly different central limit conjecture, which agrees in principle with the observations in [10], but additionally has theoretical foundations in the classical occupancy problem for urn models (e.g., see [7]).

Common Words and Analysis of Pattern Matching Algorithms Using q-gram Filtration. The NCW statistic has many applications. It serves as a distance measure between texts, especially for the comparison of biological sequences [11, 17, 13, 6]. It is also important for the analysis of text algorithms and especially for pattern matching algorithms. Here we consider the analysis of filtration algorithms for the *k-difference approximate pattern matching problem* defined as follows: Given a pattern P and a text T, find all ending positions i

in T of an approximate match of P with at most k differences (substitutions, insertions or deletions). Filtration algorithms use a quickly verifiable necessary condition for a match to filter out parts of T where no match can occur; the remaining parts are checked with a slower dynamic programming algorithm. Several filtration strategies are based on the condition that an approximate match and P should share a sufficient number of q-grams (among others, see [8, 18, 19]). Algorithms based on q-gram filtration perform very well in practice when the filtration condition is not fulfilled very often. The average running time depends on the NCW. The ability to compute its expectation should allow to analyze the average time complexity and to determine under which conditions which algorithm performs best. We were motivated to examine word statistics in random texts since we hope to analyze and fine-tune the recently proposed QUASAR algorithm [2], which uses the q-gram filtration approach. It serves to search for similar biological sequences in large databases, and has has been shown to be an order of magnitude faster than standard programs like BLAST [1].

Organization of this article. As of today, the literature does not report any exact systematic statistical study of the NMW and NCW in random texts (but see [10, 12] for some results). We describe an efficient method to calculate the exact expectations in Section 2. They depend on the probability that a q-gram does not occur in a text of length n, which itself depends only on the *autocorrelation* of the q-gram. The autocorrelation is a binary vector giving the shifts that cause the word to overlap itself. Our method requires the computation of all possible autocorrelations of length q, for which we propose the first efficient algorithm (Section 3).

The difficulty in computing the exact expectations arises from the fact that successive q-grams in a text overlap and are hence dependent. Treating them as if they were independent, one obtains the so-called classical occupancy problem for urn models (see [7] or Section 4). In this much simpler setup, many results are known, and we propose to use them as approximations to the missing words problem. The quality of these approximations is evaluated in Section 5, where we also give some exemplary results on the two applications mentioned above.

2 Computation of Expectations

2.1 Expected Number of Missing q-grams

We assume that the word length $q \geq 2$ and the alphabet size $\sigma \geq 2$ are fixed. We denote by $X^{(n)}$ the random number of missing q-grams in a text $T^{(n)}$ of length n. Assume we have enumerated the σ^q possible q-grams in some arbitrary order; let us call them $Q_1, Q_2, \ldots, Q_{\sigma^q}$. For the event that a q-gram Q occurs in $T^{(n)}$ we use the shorthand notation $Q \in T^{(n)}$, and $Q \notin T^{(n)}$ for the complementary event. By the method of indicator variables we find that

$$\mathbb{E}[X^{(n)}] = \sum_{i=1}^{\sigma^q} \Pr(Q_i \notin T^{(n)}). \tag{1}$$

An important point is that even in the symmetric Bernoulli model the probabilities of non-occurrence are not the same for every q-gram. For example, among the eight bit-strings of length 3, five do not contain the 2-gram '00', while four do not contain the 2-gram '01'. The probability depends on the autocorrelation of the q-gram, defined as follows (see also [4, 16]).

Definition 1 (Autocorrelation of a q-gram). Let $Q = Q[0] \ldots Q[q-1]$ be a q-gram over some alphabet. Then we define its *autocorrelation* $c(Q) \equiv c :=$ $(c_0, \ldots, c_{q-1})$ as follows: For $i = 0, \ldots, q-1$, set $c_i := 1$ iff the pattern overlaps itself if slided i positions to the right, i.e., iff $Q[i+j] = Q[j]$ for all $j = 0, \ldots, (q-i-1)$. Otherwise, set $c_i := 0$. Note that by this definition we always have $c_0 = 1$.

The corresponding *autocorrelation polynomial* $(C(Q))(z) \equiv C(z)$ is obtained by taking the bits as coefficients: $C(z) := c_0 + c_1 z + c_2 z^2 + \cdots + c_{q-1} z^{q-1}$.

As an example, consider the 11-gram $Q=$ABRACADABRA. By looking at the positions where this word can overlap itself, we find that $c(Q) = (10000001001)$, and therefore $(C(Q))(z) = 1 + z^7 + z^{10}$.

The following lemma gives the probability of the event $Q \notin T^{(n)}$.

Lemma 1 (Guibas & Odlyzko [5]). Let Q be a q-gram over some alphabet of size σ; let $C(z)$ be its autocorrelation polynomial. Then the generating function $P(z)$ of the sequence $(p_n := \Pr(Q \notin T^{(n)}))_{n \geq 1}$ is given by

$$P(z) = \frac{C\left(\frac{z}{\sigma}\right)}{\left(\frac{z}{\sigma}\right)^q + (1-z) \cdot C\left(\frac{z}{\sigma}\right)}.$$

The lemma states that, if we expand $P(z)$ as a power series $P(z) = \sum_{n=0}^{\infty} p_n z^n$, then the coefficient p_n of z^n is exactly $\Pr(Q \notin T^{(n)})$. We will use the usual "coefficient extraction" notation $[z^n]P(z)$ to refer to p_n.

Efficient computation. In the symmetric Bernoulli model, the probability of non-occurrence of a q-gram depends only on the alphabet size and on the autocorrelation, but not on the q-gram itself. Thus, we can simplify the calculation of $\mathbb{E}[X^{(n)}]$ by grouping together q-grams with the same autocorrelation. Let us enumerate the distinct autocorrelations that occur among the q-grams in some arbitrary order: $C_1(z), C_2(z), \ldots, C_\kappa(z)$, with κ being their number. Let $P_j(z)$ be the generating function of Lemma 1 with $C(z)$ replaced by $C_j(z)$, and let N_j be the number of q-grams with autocorrelation polynomial $C_j(z)$. We refer to N_j as the *population size* of the autocorrelation $C_j(z)$. With these definitions and Lemma 1, Equation (1) becomes

$$\mathbb{E}[X^{(n)}] = \sum_{j=1}^{\kappa} N_j \cdot [z^n]P_j(z). \tag{2}$$

To evaluate this sum, we must know the κ different polynomials $C_j(z)$ and their population sizes N_j. This is discussed in Section 3. First, we mention how to evaluate $[z^n]P_j(z)$ efficiently.

2.2 Coefficient Extraction

Note that the $P_j(z)$ $(j = 1, \ldots, \kappa)$ are rational functions of the form $\frac{f(z)}{g(z)}$, where f and g are polynomials in z such that $\deg(f) < \deg(g)$ (here $\deg(f) = q - 1$ and $\deg(g) = q$). Moreover, f and g have no common factors, and $g(0) = 1$. For the remainder of this section, call a function of this form a *standard* rational function.

There is an exact method to compute the coefficients of the power series expansion of standard rational functions in time $O(q^3 \log n)$, e.g., see [9].

However, in our case the generating functions P_j are particularly well behaved when it comes to asymptotic approximation based on the following lemma.

Lemma 2 (Asymptotic Approximation). Let $P(z) = \frac{f(z)}{g(z)}$ be a standard rational function, and let $b_1, \ldots, b_q$ be the (not necessarily distinct) poles of $P(z)$, i.e., the complex values for which $g(z) = 0$, appearing according to their multiplicity and ordered such that $|b_1| \leq |b_2| \leq \ldots \leq |b_q|$. If $|b_1| < |b_2|$, then $\beta := b_1 \in \mathbb{R}$ and

$$p_n \quad \sim \quad -\frac{f(\beta)}{g'(\beta)} \cdot \beta^{-(n+1)},$$

where $\sim$ means that the ratio of the left and right hand sides tends to 1 as $n \to \infty$. The error drops roughly as $\left| \frac{b_1}{b_2} \right|^n$.

Proof. See [16], or any textbook about linear difference equations. □

To apply the lemma, we need to verify that the $P_j(z)$ fulfill the $|b_1| < |b_2|$ condition. The rather technical proof is not included here; a proof along similar lines can be found in [3]. It turns out that the pole of smallest absolute value of $P_j(z)$ is always slightly larger than 1, whereas the absolute values of the other poles are much larger. Hence $|b_1|/|b_2|$ is small, and the asymptotic approximation is already excellent for small values of n, such as $n = 2q$. A good way to determine $\beta = b_1$ is Newton's method with a starting value of 1. We summarize the main result:

Theorem 1 (Expected Number of Missing Words). Except for a negligible asymptotic approximation error,

$$\mathbb{E}[X^{(n)}] = \sum_{j=1}^{\kappa} -N_j \cdot \frac{f_j(\beta_j)}{g_j'(\beta_j)} \cdot \beta_j^{-(n+1)},$$

where $f_j(z) := C_j(z/\sigma)$ and $g_j(z) := (z/\sigma)^q + (1 - z) \cdot C_j(z/\sigma)$, and β_j is the unique root of smallest absolute value of $g_j(z)$. □

2.3 Expected Number of Common q-grams of Two Texts

The same principles can be used to compute the expected number of q-grams that are common to two independent random texts $T^{(n)}$ and $S^{(m)}$ of lengths n and m, respectively.

We need to define more precisely how to count common q-grams. If a given fixed q-gram Q appears $x > 0$ times in $T^{(n)}$ and $y > 0$ times in $S^{(m)}$, should that count as (a) only 1, (b) x, (c) y, or (d) xy common q-gram(s)? The answer depends on the counting algorithm. Case (a) suggests that we check for each of the σ^q q-grams whether it occurs in both texts and increment a counter if it does. Case (b) implies that we look at each q-gram in $T^{(n)}$ successively, and increment a counter by one if we find the current q-gram also in $S^{(m)}$. Case (c) is symmetric to (b), and case (d) suggests that for each q-gram of $T^{(n)}$, a counter is incremented by the number of times the q-gram appears in $S^{(m)}$.

Let $Z_\bullet^{(n,m)}$ denote the number of q-grams that occur simultaneously in both $T^{(n)}$ and $S^{(m)}$ according to case $(\bullet)$ with $\bullet \in \{$a,b,c,d$\}$.

Theorem 2 (Expected Number of Common Words). Except for a negligible asymptotic approximation error,

$$\mathbb{E}[Z_a^{(n,m)}] = \sum_{j=1}^{\kappa} N_j \cdot \left(1 + \frac{f_j(\beta_j)}{g_j'(\beta_j)} \beta_j^{-(n+1)}\right) \cdot \left(1 + \frac{f_j(\beta_j)}{g_j'(\beta_j)} \beta_j^{-(m+1)}\right)$$

$$\mathbb{E}[Z_b^{(n,m)}] = \frac{n-q+1}{\sigma^q} \left(\sigma^q + \sum_{j=1}^{\kappa} N_j \frac{f_j(\beta_j)}{g_j'(\beta_j)} \beta_j^{-(m+1)}\right)$$

$$\mathbb{E}[Z_d^{(n,m)}] = \frac{(n-q+1) \cdot (m-q+1)}{\sigma^q} \quad \text{(This is exact.)}$$

The result for $\mathbb{E}[Z_c^{(n,m)}]$ is obtained by exchanging n and m in the result for $\mathbb{E}[Z_b^{(n,m)}]$. We have used κ, N_j, $f_j(z)$, $g_j(z)$ and β_j as in Theorem 1.

Proof. As before, let $Q_1, \ldots, Q_{\sigma^q}$ be an arbitrary enumeration of all q-grams over Σ. For case (a), we have $\mathbb{E}[Z_a^{(n,m)}] = \sum_{i=1}^{\sigma^q} \Pr(Q_i \in T^{(n)} \text{ and } Q_i \in S^{(m)})$. By using the independence of the texts, noting $\Pr(Q_i \in T^{(n)}) = 1 - \Pr(Q_i \notin T^{(n)})$, and grouping q-grams with the same autocorrelation together, we obtain

$$\mathbb{E}[Z_a^{(n,m)}] = \sum_{j=1}^{\kappa} N_j \cdot (1 - [z^n]P_j(z)) \cdot (1 - [z^m]P_j(z)).$$

An application of Lemma 2 proves the theorem for case (a).

For case (b), let $Y^{(m)}$ be the number of missing words in $S^{(m)}$. Then the expected number of different q-grams appearing in $S^{(m)}$ is $\sigma^q - \mathbb{E}[Y^{(m)}]$. Each of these is expected to appear $(n-q+1)/\sigma^q$ times in $T^{(n)}$. Since the texts are independent, we obtain $\mathbb{E}[Z_b^{(n,m)}] = \frac{n-q+1}{\sigma^q}(\sigma^q - \mathbb{E}[Y^{(m)}])$, and an application of Theorem 1 proves case (b).

Case (c) is symmetric to case (b), and case (d) is left to the reader. □

3 Enumerating Autocorrelations and Population Sizes

Enumerating the Autocorrelations. Our enumeration algorithm is based on a recursively defined test procedure $\Xi(v)$ proposed by Guibas and Odlyzko in

[4]. It takes as input a binary vector v of arbitrary length q, and outputs 'true' if v arises as the autocorrelation of some q-gram over any alphabet[1], and 'false' otherwise.

To enumerate $\Gamma(q)$, the set of all autocorrelations of length q, one could in principle use Ξ to test every binary vector $v = (v_0, \ldots, v_{q-1})$ with $v_0 = 1$ whether it is an autocorrelation, but this requires an exponential number of tests. The recursive structure of $\Xi(v)$ allows to build an efficient dynamic programming algorithm for the enumeration of $\Gamma(q)$, which we describe now.

First, a notational remark. While we use n for the text length in the other sections, here we use n as a running variable for the autocorrelation length, $n = 1, \ldots, q$. This should cause no confusion.

For a binary vector v of length n with $v_0 = 1$, define $\pi(v) := \min\{1 \le i < n : v_i = 1\}$, and let $\pi(v) := n$ if no such i exists. For $n \ge 1$, $p = 1, \ldots, n$, let $\Gamma(n, p)$ be the set of autocorrelations v of length n for which $\pi(v) = p$. Then $\Gamma(n) = \bigcup_{p=1}^{n} \Gamma(n, p)$. We denote the concatenation of two bit vectors s and t by $s \circ t$, and the k-fold concatenation of s with itself by s^k. So $10^k \circ w$ is the binary vector starting with 1, followed by k 0s, and ending with the binary vector w.

From the definition of autocorrelation we have that $v \in \Gamma(n, p)$ implies $v_{jp} = 1$ for all $j = 1, 2, \ldots$, for which $jp < n$. It follows that $\Gamma(n, 1) = \{1^n\}$. Also, $\Gamma(n, n) = \{10^{n-1}\}$. To obtain $\Gamma(n, p)$ for $n \ge 3$ and $2 \le p \le (n-1)$, we assume all $\Gamma(m, p')$ with $m < n$, $1 \le p' \le m$, are already known. Then there are two cases:

Case (a) $[2 \le p \le \frac{n}{2}]$: Let $r' := n \bmod p$ and $r := r' + p$. Then $p \le r < 2p$, and there are two sub-cases. In each of them, $\Gamma(n, p)$ can be constructed from a subset of $\Gamma(r)$. Hence, let $s_{n,p} := (10^{p-1})^{\lfloor n/p \rfloor - 1}$; every string in $\Gamma(n, p)$ has $s_{n,p}$ as a prefix, and is then followed by a string $w \in \Gamma(r)$, as follows.

1. Case $r = p$:

$$\Gamma(n, p) = \{s_{n,p} \circ w \mid w \in \Gamma(r, p'); \ r' + \gcd(p, p') < p' < p\} \qquad (3)$$

2. Case $p < r < 2p$:

$$\Gamma(n, p) = \{s_{n,p} \circ w \mid w \in \Gamma(r, p)\} \qquad (4)$$
$$\dot{\bigcup} \{s_{n,p} \circ w \mid w \in \Gamma(r, p'); \ r' + \gcd(p, p') < p' < p; \ w_p = 1\}$$

We remark that the condition imposed on p' in (3) and (4) $(r' + \gcd(p, p') < p' < p)$ implies that p' must not divide p.

Case (b) $[\frac{n}{2} < p \le (n-1)]$: $\Gamma(n, p)$ is constructed from $\Gamma(n - p)$.

$$\Gamma(n, p) = \{10^{p-1} \circ w \mid w \in \Gamma(n - p)\} \qquad (5)$$

---

[1] In [4], it is proved that if v is an autocorrelation of some word over an alphabet of size $\sigma \ge 2$, then it is also the correlation of word over a two-letter alphabet.

The equations (3), (4) and (5), along with the known forms of $\Gamma(n, 1)$ and $\Gamma(n, n)$ for all n, yield a dynamic programming algorithm for the construction of $\Gamma(q)$. For $n = 3, \ldots, q$, in order to compute all $\Gamma(n, p)$ with $1 \leq p \leq n$, one needs the sets of autocorrelations of shorter lengths $\Gamma(m, p')$ with $m < \lfloor \frac{2n}{3} \rfloor$, $1 \leq p' \leq m$. The correctness of the algorithm follows from the correctness of the recursive predicate $\Xi(v)$ in [4], after which it is modeled.

One improvement is possible: In case (a), $\Gamma(n, p)$ is obtained from auto-correlations $w \in \Gamma(r)$ with $r \geq p$. Examining the characterization of auto-correlations given in [4] more closely, it can be shown that such w must have $\pi(w) > (n \bmod p)$, and therefore it is possible to construct $\Gamma(n, p)$ from the sets $\Gamma(s)$ with $s < p$. Hence, to obtain $\Gamma(n, p)$, in both cases (a) and (b), only the sets $\Gamma(m, p')$ with $m \leq \lfloor \frac{n}{2} \rfloor$, $1 \leq p' \leq m$ are needed. For example, to compute $\Gamma(200)$, we only need to know $\Gamma(1), \ldots, \Gamma(100)$, but not $\Gamma(101), \ldots, \Gamma(199)$.

The Number of Autocorrelations. When we know $\Gamma(q)$, we also know its cardinality $\kappa(q)$, which is the number of terms in the sum of (2). In [4] it is proved that asymptotically

$$\kappa(q) \leq \exp\left(\frac{(\ln q)^2}{2\ln(3/2)} + o((\ln q)^2) \right) ;$$

hence computing the expected number of missing words by (2) is considerably more efficient than by (1), which uses σ^q terms.

Population sizes. To determine how many q-grams over an alphabet of size σ share a given autocorrelation $v \in \Gamma(q)$ (the population size $N(v)$), we refer the reader to [4, Section 7], where a recurrence for $N(v) = N((v_0, \ldots, v_{q-1}))$ in terms of $N((v_{\pi(v)}, \ldots, v_{q-1}))$ is given, or to our technical report [15], where an alternative method is described.

4 Approximations

Since for large q, the exact methods in Section 2 become very time-consuming, we consider a simpler but related problem, whose solution we take as an approximate solution to the missing words problem.

Classical Occupancy Problem: When N balls are independently thrown into M equiprobable urns, what is the distribution of the number of empty urns X after the experiment? For this setup, the moments of X (expectation, variance, and higher moments) are known. For example, we have that

$$\mathbb{E}[X] = M\left(1 - \frac{1}{M}\right)^N , \tag{6}$$

$$\mathrm{Var}[X] = M\left(1 - \frac{1}{M}\right)^N + M(M-1)\left(1 - \frac{2}{M}\right)^N - M^2\left(1 - \frac{1}{M}\right)^{2N} \tag{7}$$

Even the distribution of X can be given explicitly in terms of the so-called Stirling numbers of the second kind. From this knowledge, the following result can be derived.

Lemma 3 (Central Limit Theorem for the Classical Occupancy Problem). Let (N_k) and (M_k) be sequences of natural numbers such that $N_k \to \infty$, $M_k \to \infty$ and $\frac{N_k}{M_k} \to \lambda > 0$, as $k \to \infty$. Let (X_k) be the sequence of random variables denoting the number of empty urns after N_k balls have been thrown independently into M_k urns. Then, as $k \to \infty$, we have

$$\mathbb{E}[X_k/M_k] \to e^{-\lambda}, \tag{8}$$

$$\mathrm{Var}[X_k/\sqrt{M_k}] \to (e^\lambda - 1 - \lambda)e^{-2\lambda}. \tag{9}$$

Moreover, we have convergence in distribution

$$\frac{X_k - M_k \, e^{-\lambda}}{\sqrt{M_k \, (e^\lambda - 1 - \lambda) \, e^{-2\lambda}}} \;\xrightarrow{\mathcal{D}}\; \mathcal{N}, \tag{10}$$

where $\mathcal{N}$ denotes the standard normal distribution.

Proof. See, for example, [7]. $\qquad\square$

The missing words problem is a modification of the classical occupancy problem in the following sense. The M urns correspond to the σ^q possible words, and the N balls correspond to the $(n - q + 1)$ q-grams of the text. The difference is that successive q-grams in the text are strongly dependent because they overlap by $(q - 1)$ characters, while the balls in the occupancy problem are assumed to be independent.

Approximations. We propose to use the results from the occupancy problem by assuming that the q-grams are not taken from a text but generated independently. Then the probability that a fixed q-gram does not occur among $n - q + 1$ randomly drawn q-grams is $(1 - \frac{1}{\sigma^q})^{n-q+1}$. Hence, the expected number of missing words can be approximated by (6), giving

$$\mathbb{E}[X^{(n)}] \approx \sigma^q \cdot \left(1 - \frac{1}{\sigma^q}\right)^{n-q+1} \tag{11}$$

This can be further approximated by using the asymptotic value from (8), resulting in

$$\mathbb{E}[X^{(n)}] \approx \sigma^q \cdot e^{-\lambda}, \qquad \lambda := \frac{n - q + 1}{\sigma^q} \tag{12}$$

We call (11) the *independence approximation* (IA), and (12) the *exponential approximation* (EA).

A Central Limit Conjecture for Missing Words. The numerical results in Section 5 show that the independence approximation is surprisingly good. Also, the asymptotic variance in the occupancy problem $(M_k \, (e^\lambda - 1 - \lambda) \, e^{-2\lambda})$ is in accordance with the variance of the number of missing words observed by Marsaglia & Zaman in [10] $(M_k \, (e^\lambda - 3) \, e^{-2\lambda})$ when they conducted simulations for slightly varying $\lambda \approx 2$. Although we have not checked higher moments, we formulate the following conjecture.

Conjecture 1. The number of missing q-grams over an alphabet of size σ in a text of length n, and the number of empty urns after $N := n - q + 1$ balls have been independently thrown into $M := \sigma^q$ urns have the same Gaussian limit distribution, as given in Lemma 3.

Further Remarks. Other approximations are possible. For example, one may assume that the waiting time until a q-gram first occurs in the text has a geometric distribution. Due to space constraints, we do not consider this approximation any further here. In our tests, (IA) always turned out to be better by an order of magnitude. Also, (IA) and (EA) can be applied to the common words problem as well, as shown here for case (b) from Section 2.3:

(IA): $\mathbb{E}[Z_b^{(n,m)}] \approx (n - q + 1) \cdot \left(1 - \left(1 - \frac{1}{\sigma^q}\right)^{m-q+1}\right)$

(EA): $\mathbb{E}[Z_b^{(n,m)}] \approx (n - q + 1) \cdot \left(1 - \exp\left(\frac{m-q+1}{\sigma^q}\right)\right)$

5 Comparison of the Methods

We evaluate the quality of the approximations in two different scenarios of practical relevance.

A Monkey Test of High Order. Coming back to the monkey test described in Section 1, we compute the expected number of missing bit-strings of length 33 in a random text of varying length n exactly by Theorem 1 (XT), and approximately according to the independence approximation and the exponential approximation from Section 4. The excellent quality of the approximations can be seen in Table 1. The relative errors are around 10^{-8} or lower. Since for the exact method, $\kappa = \kappa(33) = 538$ different correlations have to be evaluated, the approximations save time without sacrificing quality. This behavior is quite typical when the alphabet size, the word length and the text length are sufficiently large. For some further numerical results, see [14]. For all computations, we used 100-digit-precision arithmetic. High precision is necessary for the exact computation and for (IA), as the standard floating point introduces large roundoff errors. (EA) is more robust in this respect.

QUASAR Analysis. The QUASAR algorithm [2] can be used to search DNA databases (long texts with $\sigma = 4$) for approximate matches to a pattern of length $n = 50$ with at most $k = 7$ differences. Assume that the database has been partitioned into blocks of size $m = 256$. Assume further that an index is available to quickly determine how many q-grams $Z \equiv Z_b^{(n,m)}$ each block and the pattern have in common according to Section 2.3, Case (b), for $q \in \{3, 4, 5, 6\}$. By a lemma of Jokinen & Ukkonen [8], no k-approximate match can occur in a block if the number of common q-grams is below $t(q) := n - q + 1 - kq$. Hence in the event $Z \geq t(q)$ a block must be kept for further inspection. It is natural to ask for which value of q the filtration performance is optimal, i.e., $\Pr(Z \geq t(q))$ is minimal. When only $\mathbb{E}[Z]$ is known, $\Pr(Z \geq t(q))$ can be bounded with Markov's inequality, $P(|Z| \geq t) \leq \mathbb{E}[|Z|]/t$. Table 2 lists some relevant values. The approximation error is of the order 10^{-3}, i.e., approximately 0.1%.

Table 1. Expected number of missing 33-grams for alphabet size $\sigma = 2$ and varying text length n. Line 1 (XT) shows the exact values according to Theorem 1. Line 2 (IA) gives the independence approximation (IA), and Line 3 the order of magnitude of its relative error. Line 4 shows the exponential approximation (EA), and Line 5 the order of magnitude of its relative error.

	Expected Fraction of Missing q-grams, $\mathbb{E}[X^{(n)}]/\sigma^q$, for $\sigma = 2$ and $q=33$			
	Method	$n = 0.5\,\sigma^q$	$n = \sigma^q$	$n = 2\,\sigma^q$	$n = 4\,\sigma^q$
1	(XT)	0.606530661913	0.36787944248	0.135335283715	0.018315638966
2	(IA)	0.606530661954	0.36787944252	0.135335283725	0.018315638952
3	$\log_{10}$(Rel.Error)	-10.169	-9.972	-10.147	-9.124
4	(EA)	0.606530661972	0.36787944254	0.135335283741	0.018315638957
5	$\log_{10}$(Rel.Error)	-10.014	-9.783	-9.726	-9.285

Table 2. Expected number of common q-grams of two texts of length 50 and 256 according to case (b) of Section 2.3, for varying q. The alphabet size is $\sigma = 4$. Line 1 shows the exact value (XT). Line 2 gives the independence approximation (IA), and Line 3 shows the magnitude of its relative error. Line 4 gives the exponential approximation (EA), and Line 5 the order of its relative error. Line 6 shows the Jokinen-Ukkonen threshold $t(q)$ for q-gram filtration to find matches with $k = 7$ differences. Line 7 gives an upper bound on the probability of at least $t(q)$ common q-grams, based on Markov's inequality.

	Quantity	$q = 3$	$q = 4$	$q = 5$	$q = 6$
1	$\mathbb{E}[Z]$ (XT)	47.10188	29.55391	10.04229	2.67534
2	$\mathbb{E}[Z]$ (IA)	47.12092	29.53968	10.03927	2.67509
3	$\log_{10}$(Rel.Error(IA))	-3.393	-3.317	-3.523	-4.041
4	$\mathbb{E}[Z]$ (EA)	47.09294	29.50585	10.03495	2.67478
5	$\log_{10}$(Rel.Error(EA))	-3.722	-2.789	-3.136	-3.679
6	$t(q)$	27	19	11	3
7	$\Pr(Z \geq t(q))$ (Markov)	trivial ≤ 1	trivial ≤ 1	≤ 0.913	≤ 0.892

If one is willing to accept the simple random input model[2], $q = 6$ offers the best bound, but a more precise statement can only be made if more is known about the distribution of Z.

6 Conclusion and Future Work

We have presented exact and approximate methods to compute the expected number of missing q-grams in a random text and the expected number of common words in two independent random texts. The exact computations require the

---

[2] DNA is surely not a "random text" since it contains biological information. Still, the analysis of the algorithm under a simple random input model can give valuable hints to its performance on real data, as long as no stronger formal assumptions about the data can be made.

knowledge of all autocorrelations of length q, for which we exhibit an efficient dynamic programming algorithm.

We observe that in general the independence approximation (IA) gives excellent results, although clearly the q-grams of a text are not independent. An explanation is that, although for each i, $\Pr(Q_i \notin T^{(n)})$ may be very different from $(1 - \frac{1}{\sigma^q})^n$, there is an averaging effect in the sum over all q-grams, such that (IA) continues to hold approximately. We have been unable to prove this, though. It also remains an open problem to prove or disprove Conjecture 1, or to make it more precise. We wanted to point out the relatedness of the NCW problem and the classical occupancy problem, which seems to have been overlooked in related work.

For short texts, small alphabets and small word lengths, the errors due to the use of either approximation are quite high. The exact method is then advised, since it is inexpensive in these cases. However, high precision arithmetic should be used to avoid roundoff errors. In the other cases, (EA) is the most reasonable choice, because its evaluation poses the least numerical problems, and the approximation error can be neglected.

References

[1] S. Altschul, W. Gish, W. Miller, E. Myers, and D. Lipman. Basic Local Alignment Search Tool (BLAST). *Journal of Molecular Biology*, 215:403–410, 1990.

[2] S. Burkhardt, A. Crauser, P. Ferragina, H.-P. Lenhof, E. Rivals, and M. Vingron. q-gram based Database Searching Using a Suffix Array (QUASAR). In S. Istrail, P. Pevzner, and M. Waterman, editors, *Proceedings of The Third International Conference on Computational Molecular Biology*, pages 77–83. ACM-Press, 1999.

[3] L. J. Guibas and A. M. Odlyzko. Maximal Prefix-Synchronized Codes. *SIAM Journal of Applied Mathematics*, 35(2):401–418, 1981.

[4] L. J. Guibas and A. M. Odlyzko. Periods in Strings. *Journal of Combinatorial Theory, Series A*, 30:19–42, 1981.

[5] L. J. Guibas and A. M. Odlyzko. String Overlaps, Pattern Matching, and Non-transitive Games. *Journal of Combinatorial Theory, Series A*, 30:183–208, 1981.

[6] W. Hide, J. Burke, and D. Davison. Biological evaluation of d2, an algorithm for high-performance sequence comparison. *J. Comp. Biol.*, 1:199–215, 1994.

[7] N. L. Johnson and S. Kotz. *Urn Models and Their Applications*. Wiley, New York, 1977.

[8] P. Jokinen and E. Ukkonen. Two algorithms for approximate string matching in static texts. In A. Tarlecki, editor, *Proceedings of the 16th symposium on Mathematical Foundations of Computer Science*, number 520 in Lecture Notes in Computer Science, pages 240–248, Berlin, 1991. Springer-Verlag.

[9] D. E. Knuth. *The Art of Computer Programming*, volume 2 / Seminumerical Algorithms. Addison-Wesley, Reading, MA, third edition, 1998.

[10] G. Marsaglia and A. Zaman. Monkey Tests for Random Number Generators. *Computers and Mathematics with Applications*, 26(9):1–10, 1993.

[11] A. A. Mironov and N. N. Alexandrov. Statistical method for rapid homology search. *Nucleic Acids Res*, 16(11):5169–73, Jun 1988.

[12] O. E. Percus and P. A. Whitlock. Theory and Application of Marsaglia's Monkey Test for Pseudorandom Number Generators. *ACM Transactions on Modeling and Computer Simulation*, 5(2):87–100, April 1995.

[13] P. A. Pevzner. Statistical distance between texts and filtration methods in sequence comparison. *Comp. Appl. BioSci.*, 8(2):121–127, 1992.

[14] S. Rahmann and E. Rivals. The Expected Number of Missing Words in a Random Text. Technical Report 99-229, LIRMM, Montpellier, France, 1999.

[15] E. Rivals and S. Rahmann. Enumerating String Autocorrelations and Computing their Population Sizes. Technical Report 99-297, LIRMM, Montpellier, France, 1999.

[16] R. Sedgewick and P. Flajolet. *Analysis of Algorithms.* Addison-Wesley, Reading, MA, 1996.

[17] D. C. Torney, C. Burks, D. Davison, and K. M. Sirotkin. Computation of d^2: A measure of sequence dissimilarity. In G. Bell and R. Marr, editors, *Computers and DNA*, pages 109–125, New York, 1990. Sante Fe Institute studies in the sciences of complexity, vol. VII, Addison–Wesley.

[18] E. Ukkonen. Approximate string-matching with q-grams and maximal matches. *Theoretical Computer Science*, 92(1):191–211, Jan. 1992.

[19] S. Wu and U. Manber. Fast text searching allowing errors. *Communications of the Association for Computing Machinery*, 35(10):83–91, Oct. 1992.