



ACADEMIC
PRESS

Available at
www.ComputerScienceWeb.com
POWERED BY SCIENCE @ DIRECT®

Journal of Computer and System Sciences 68 (2004) 22–40

JOURNAL OF
COMPUTER
AND SYSTEM
SCIENCES

<http://www.elsevier.com/locate/jcss>

Sensitivity analysis and efficient method for identifying optimal spaced seeds[☆]

Kwok Pui Choi^{a,b} and Louxin Zhang^{c,*}

^a *Department of Mathematics, National University of Singapore, Singapore 117543, Singapore*

^b *Department of Statistics and Applied Probability, National University of Singapore, Singapore 117543, Singapore*

^c *Department of Mathematics, National University of Singapore, 2 Science Drive 2, Singapore 117543, Singapore*

Received 4 February 2003; revised 24 April 2003

Abstract

The novel introduction of spaced seed idea in the filtration stage of sequence comparison by Ma et al. (Bioinformatics 18 (2002) 440) has greatly increased the sensitivity of homology search without compromising the speed of search. Finding the optimal spaced seeds is of great importance both theoretically and in designing better search tool for sequence comparison. In this paper, we study the computational aspects of calculating the hitting probability of spaced seeds; and based on these results, we propose an efficient algorithm for identifying optimal spaced seeds.

© 2003 Elsevier Inc. All rights reserved.

Keywords: Sequence comparison; Pattern matching; Filtration technique; Spaced seeds; Sensitivity analysis; Heuristic algorithm

1. Introduction

With more and more genomes being completely sequenced, comparison of multimegabase genomic DNA sequences has become an important technique for genome annotation. By comparing orthologous genomic sequences, information on SNPs, translocation, tandem and segmental duplications, and intronic and intergenic regions with potential biological function can easily be inferred [8,12,16]. This unprecedented demand for long genomic DNA sequence comparisons poses a great challenge to alignment algorithm developers, as the popular programs

[☆]This work is partially supported by BMRC Research Grant BMRC01/1/21/19/140.

*Corresponding author. Fax: +65-6779-5452.

E-mail addresses: matckp@nus.edu.sg (K.P. Choi), matzlx@nus.edu.sg (L. Zhang).

such as FASTA, BLAST, SIM are too computationally demanding to analyze multimegabase sequences even in a modern computer [1,2,11,13,17].

One of the most important techniques for designing faster algorithms for sequence comparisons is the idea of filtration [5–7,19]. This idea involves a two-stage process. The first stage preselects a set of positions in which given sequences are potentially similar. The second stage verifies each possible position using an accurate method rejecting potential matches that do not satisfy the specified similarity criteria. For example, BLAST programs use this technique. Each of these programs first finds reasonably long exact matches (consecutive k bases) between the given sequence and a sequence in the database, and then extend these exact matches into local alignments. Based on statistical study, two sequences are likely to have high-scoring local alignments only if there are reasonably long exact matches between them. The value of k is usually set to 11 by considering tradeoff between search speed and the sensitivity. The larger the k is, the faster the program but the poorer its sensitivity.

In fact, the idea of filtration for information retrieval/pattern matching in computer science and for sequence comparison in computational molecular biology goes back almost two decades. It was first described by Karp and Rabin for the string matching problem [14]. It was also stated for alignment problems in [4,9].

Multiple spaced patterns are usually used for approximate matching and sequence comparison [5,7]. Recently, a creative idea of using a single optimal spaced pattern (called spaced seed) was introduced in designing a more efficient and sensitive program PatternHunter for sequence comparison [18]. PatternHunter also considered multiple hits of same or different seeds in the filtration stage. According to their results, the optimal spaced seed of weight 11 and length 18 is as sensitive as a consecutive seed of weight 10 when both compared sequences have 70% similarity and produces 4 times fewer chance hits. They further demonstrated that the running time of PatternHunter in a personal computer is seconds for searching bacterial genomes, minutes for Arabidopsis chromosomes, and hours for human chromosomes.

These significant practical applications raise the following questions about spaced seeds. Given the similarity of compared sequences,

- (i) do all spaced seeds have greater sensitivity than the consecutive seed of the same weight?
- (ii) how does one find the optimal spaced seeds of a given weight? (Optimal in the sense of greatest sensitivity.)

Obviously, the answers to these questions are critical for better designs, and justifying the filtration techniques in existing applications. They may also open up new ways for exploring various applications in sequence comparison.

In some aspects, all spaced seeds have better performance than the consecutive seed. For example, Keich et al. proved that the expected value of the first hitting time of any spaced seed is strictly less than that of the consecutive one of the same weight [15]. However, this beautiful result has not demonstrated that the consecutive seed has the least sensitivity. Indeed, the answer to question (i) is no. There are spaced seeds which are less sensitive than the consecutive seed of the same weight (c.f. the paragraph after the proof of Proposition 3.1). This calls for a judicious choice of spaced seed. However, giving a comprehensive/satisfactory answer to question (ii) turns out to be extremely difficult. Two dynamical programming algorithms are given in [15] to search

for the optimal spaced seeds for a given weight w and given length L . However, the time complexity of their algorithms grow exponentially fast with $L - w$ and linearly with L . These algorithms are too computationally demanding to identify the optimal seeds of weights larger than 13 in a PC in a reasonably short time, which is not sufficient for designing faster tools for sequence comparison. (Note that MegaBLAST uses a consecutive seed of weight as big as 28.) Is there a faster way to do so? Run statistics have been studied for several decades (see [10,22,23]; just to name a few), and one suspects that results there might help here. However, the theory is not directly applicable to question (ii). In Section 2, we are able to derive sets of recurrence relations for computing the hit probabilities of spaced seeds. And we answer some questions for the hitting probabilities from theoretical and computational points of view. Armed with these recurrent relations, we have done extensive numerical calculations for hit probabilities and come up with an efficient screening algorithm for identifying optimal spaced seeds of the given weight and length. This algorithm turns out to be at least 10 times faster than the existing ones. Although our approach is heuristic, the program consistently produces the optimal spaced seeds for our test cases with medium weight and length.

Without any doubt, our approach for identifying optimal spaced seeds will greatly facilitate designing better programs for homology search and analysis of multimegabase sequences. Furthermore, it is hopeful that our work may find applications in other bioinformatics research such as sequencing by hybridization where spaced probes schemes were proposed by Preparata et al. [20,21].

2. Computing the sensitivity of a seed

As mentioned before, the BLAST program looks for a short substring of length k that appears in both the query and target sequences in the filtration stage. But, Ma et al. empirically discovered that better filter can be obtained if spaced k positions are examined [18]. Such a pattern of the matching positions is called a ‘seed’ in their paper. Here we denote a spaced seed by a string on $\{1, *\}$, where $*$ corresponds to the ‘don’t care’ positions. For example, if the spaced seed $Q = 1**11*1$ is used, there are two seed matches between *gcaattgccg* and *acgattgctg*; such two matches end at positions 8 and 10, respectively. Obviously, a seed can uniquely be specified by the relative positions of the 1’s in the seed [6]. The seed Q given above has the set of relative positions $\{0, 3, 4, 6\}$, where we number positions from 0. The number of 1’s in a seed is called its *weight*; its overall length is called its *length*.

A very important problem arising from Ma et al.’s work is how to find an optimal spaced seed for detecting identities in a homology region. Assume there are two DNA sequences S' and S'' of length n such that the events that S' and S'' are identical at position i (or $S'[i] = S''[i]$) are jointly independent and each event is of probability p . By translating a match at a position to 1 and mismatch to 0, we have the following equivalent formulation of the problem.

Let $S = s_1s_2\cdots$ be an infinite Bernoulli random sequence in which $S[i] := s_i$ takes only two values, 1 or 0 with probability p and $q := 1 - p$ respectively. Let Q be a spaced seed of length L and weight w given by the following relative position set:

$$\mathcal{RP}(Q) = \{i_1 = 0, i_2, \dots, i_w = L - 1\}. \quad (1)$$

The seed Q is said to hit S at position n if

$$S[n - L + i_1 + 1] = S[n - L + i_2 + 1] = \dots = S[n - L + i_w + 1] = 1,$$

where $n - L + i_w + 1 = n$. Here we use the ending position as the hitting position. Let $Q_n := Q_n(p)$ denote the probability that Q hits S before or at position n . Then, we look for an optimal seed of given weight w that has the largest hitting probability Q_n for given n and p .

2.1. A simpler formula for consecutive seeds

In this section, we consider the hitting probability of the consecutive seed B of weight w . Obviously, it is a special spaced seed without ‘don’t care’ positions. Let $B_n := B_n(p)$ denote the probability that the seed hits a random sequence S before or at position n and $\bar{B}_n = 1 - B_n, n \geq 1$. Then, $B_n = 0, n = 0, 1, \dots, w - 1$ and $B_w = p^w$. We shall derive a recursive relation of \bar{B}_n for $n \geq w + 1$. In order that B does not hit the random string $S[1, n]$ (that is, the first n positions of the random string S), the first 0 must have occurred in $S[1, w]$, therefore

$$\begin{aligned} \bar{B}_n &= \sum_{i=0}^{w-1} p^i q \bar{B}_{n-i-1} \\ &= \bar{B}_{n-1} - p \left[\bar{B}_{n-1} - \sum_{i=1}^{w-1} p^{i-1} q \bar{B}_{n-i-1} \right] \end{aligned} \tag{2}$$

$$= \bar{B}_{n-1} - p^w q \bar{B}_{n-w-1} \quad \text{using (2) for } n - 1. \tag{3}$$

Therefore, B_n can be computed in $2(n - w)$ arithmetic operations if we pre-compute the value of $p^w q$. From the formula, it is not difficult to see that

$$\begin{aligned} B_n &= p^w + (n - w)p^w q, \quad w \leq n \leq 2w, \\ B_{2w+1} &= p^w + (w + 1)p^w q - p^{2w} q. \end{aligned}$$

2.2. Formulas for spaced seeds

Computing the hitting probability of a spaced seed is much more involved. Let Q be a spaced seed of length L and weight w , which is specified by the relative position set in Formula (1). Recall that Q_n is the probability that the seed Q hits an infinite Bernoulli random sequence S before or at position n . To calculate Q_n , we let A_j be the event that Q hits S at position j and let \bar{A}_j denote the complement of A_j for any $j \leq n$. (Trivially, $A_1 = \dots = A_{L-1} = \emptyset$.) Then,

$$Q_n = P \left[\bigcup_{1 \leq i \leq n} A_i \right]$$

and the probability, f_n , that the spaced seed first hits S at position n is

$$f_n = P[\bar{A}_1 \bar{A}_2 \dots \bar{A}_{n-1} A_n].$$

Obviously,

$$Q_L = p^w = f_L, \quad Q_n = f_n = 0, \quad 1 \leq n < L,$$

where p is the probability that 1 occurs at a position in S , and

$$Q_n = Q_{n-1} + f_n, \quad n \geq 1. \tag{4}$$

Furthermore, f_n can be computed recursively as follows.

Let

$$W_Q = \{w_1, w_2, \dots, w_m\}$$

be the set of all $m := 2^{L-w}$ distinct strings w_j obtained from the seed Q by filling 1 in the ‘care’ positions i_k ($1 \leq k \leq w$), i.e. $w_j[i_k] = 1$, and 0 or 1 in the ‘don’t care’ positions. For example, for seed $Q = 1 * 1 * 1$,

$$W_Q = \{10101, 11101, 10111, 11111\}.$$

The seed Q hits at position n if and only if there is an $w_j \in W_Q$ occurs at n . For each j , we use $A_n^{(j)}$ to denote the event that the word w_j occurs at n . Then, $A_n = \bigcup_{1 \leq j \leq m} A_n^{(j)}$, and $A_n^{(j)}$ ’s are disjoint (i.e., $A_n^{(j)} A_n^{(k)} = \emptyset$ for $1 \leq j \neq k \leq m$). Setting

$$f_n^{(j)} = P[\bar{A}_1 \bar{A}_2 \cdots \bar{A}_{n-1} A_n^{(j)}], \quad 1 \leq j \leq m,$$

we have

$$f_n = \sum_{1 \leq j \leq m} f_n^{(j)}. \tag{5}$$

We use $x_j[a, b]$ to denote the substring of $x_j \in W_Q$ from position a to position b inclusively. For example, $x_j[1, L] = x_j$, where L is the length of the seed Q . Since

$$\begin{aligned} & \bar{A}_1 \bar{A}_2 \cdots \bar{A}_{n-1} A_n^{(j)} \\ &= \bar{A}_1 \bar{A}_2 \cdots \bar{A}_{n-L} A_n^{(j)} \setminus \bigcup_{i=1}^{L-1} [\bar{A}_1 \bar{A}_2 \cdots \bar{A}_{n-i-1} A_{n-i} A_n^{(j)}] \\ &= \bar{A}_1 \bar{A}_2 \cdots \bar{A}_{n-L} A_n^{(j)} \setminus \bigcup_{i=1}^{L-1} \left(\bigcup_{k=1}^m \bar{A}_1 \bar{A}_2 \cdots \bar{A}_{n-i-1} A_{n-i}^{(k)} A_n^{(j)} \right). \end{aligned}$$

and the event $A_{n-i}^{(k)} A_n^{(j)}$ implies that substrings $w_k[i + 1, L]$ and $w_j[1, L - i]$ are identical, we obtain that

$$f_n^{(j)} = (1 - Q_{n-L})P[w_j] - \sum_{i=1}^{L-1} \left(\sum_{k \in \Gamma_{ij}} f_{n-i}^{(k)} \right) P[w_j[L - i + 1, L]], \tag{6}$$

where $P[w_j]$ is the probability that the word w_j occurs at position L and

$$\Gamma_{ij} = \{k \mid w_k[i + 1, L] = w_j[1, L - i]\}.$$

Formulas (4)–(6) allow us to compute Q_n and f_n in $O(nL2^{2(L-w)})$ arithmetic operations.¹ Note that this simple recursive algorithm takes exponential time in $L - w$ and linear in n and L .

Proposition 2.1. For a spaced seed Q of length L and weight w , let b denote the number of blocks of zeros. For $i \leq j$, let $\bar{A}_{i,j} = \bar{A}_i \bar{A}_{i+1} \cdots \bar{A}_j$. Then,

$$f_n = p^w(1 - p^{b+1}) - \sum_{j=1}^{n-L-1} P[A_L \bar{A}_{L+1,L+j} A_{L+j+1}], \quad n \geq L + 2, \tag{7}$$

$$Q_n = Q_m + (n - m)f_m - \sum_{k=m}^{n-1} (n - k)P(A_L \bar{A}_{L+1,k} A_{k+1}), \quad n > m \geq L + 1. \tag{8}$$

Proof. Note that $P[A_L A_{L+1}] = p^{w+b+1}$. Observe that

$$f_n = P[\bar{A}_{L,n-1} A_n] = P[\bar{A}_{L,n-1}] - P[\bar{A}_{L,n}] = P[\bar{A}_{L+1,n}] - P[\bar{A}_{L,n}] = P[A_L \bar{A}_{L+1,n}].$$

For $n \geq L + 2$,

$$\begin{aligned} f_n &= f_L + (f_{L+1} - f_L) + \sum_{j=L+2}^n (f_j - f_{j-1}) \\ &= p^w(1 - p^{b+1}) - \sum_{j=L+2}^n P[A_L \bar{A}_{L+1,j-1} A_j] \\ &= p^w(1 - p^{b+1}) - \sum_{j=1}^{n-L-1} P[A_L \bar{A}_{L+1,L+j} A_{L+j+1}]. \end{aligned}$$

For $n > m \geq L + 1$,

$$\begin{aligned} Q_n &= Q_m + (n - m)f_m - \sum_{j=m+1}^n (f_m - f_j) \\ &= Q_m + (n - m)f_m - \sum_{j=m+1}^n \sum_{k=m}^{j-1} (f_k - f_{k+1}) \\ &= Q_m + (n - m)f_m - \sum_{j=m+1}^n \sum_{k=m}^{j-1} P[A_L \bar{A}_{L+1,k} A_{k+1}] \\ &= Q_m + (n - m)f_m - \sum_{k=m}^{n-1} (n - k)P[A_L \bar{A}_{L+1,k} A_{k+1}]. \quad \square \end{aligned}$$

¹ Multiple runs under independent trials was studied by Schwager [22]. Unfortunately, the formula for calculating the probability that multiple runs occur in the paper is incorrect.

2.3. Markov chain approach

Markov chains have been applied for studying run statistics of the consecutive seeds or compound word patterns in the past decade [10]. This method is called Markov chain embedding technique. For a comprehensive treatment, see [3] and references therein. Here we illustrate how the method is relevant in deriving a recurrence relation for the hitting probabilities of spaced seeds. We imagine that the random sequence S is revealed one bit at a time, and we are comparing this segment of the random sequence for hit. If there is no hit yet at time n , one should only keep a certain amount of the last portion of $S[1, n]$ for possible future hit, but discard the front portion of $S[1, n]$ which will definitely not be needed for future hit. A moment of thought suggests that the last portion of $S[1, n]$ to be kept must coincide with a longest prefix of strings generated by Q , the set W_Q .

More precisely, for a spaced seed Q , recall that W_Q is the set of all 2^{L-w} distinct strings obtained from Q as in Section 2.2. Let $\mathcal{P}(Q)$ denote the set of all prefixes of strings of W_Q including the empty prefix. For example, $Q = 1 * 1$, then $W_Q = \{101, 111\}$ and $\mathcal{P}(Q) = \{\epsilon, 1, 10, 11, 101, 111\}$. Here, ϵ denotes the empty string. Let the state space be denoted by $\mathcal{S} := \{x | x \in \mathcal{P}(Q), |x| < L\} \cup \{H\}$ where H stands for hit, collection of all at most $L - 1$ long prefixes from W_Q . Recall that the random bits s_1, s_2, \dots , are independent and identically distributed Bernoulli trials with probability of success p . Define $X_0 = \epsilon$, and for $n \geq 1$, we let $X_n := X_n(s_1 \cdots s_n)$ be at state H if Q hits $s_1 \cdots s_n$ at/before n ; or at state $s_{n-k+1} \cdots s_n$ where k is the smallest integer such that $s_{n-k+1} \cdots s_n \in \mathcal{S}$, i.e., the longest suffix of $S[1, n]$ that belongs to W_Q . It is not difficult to see that $\{X_0, X_1, \dots\}$ form a Markov chain starting at ϵ and its transition probabilities, denoted by $P(x, y)$, $x, y \in \mathcal{S}$, (transition probability from state x to state y) is given as follows:

For $x = H$,

$$P(x, y) = \begin{cases} 1 & \text{if } y = H, \\ 0 & \text{otherwise} \end{cases}$$

and for $x \neq H$,

$$P(x, y) = \begin{cases} p & \text{if } y = x1 \text{ or } |x| = L, \\ q & \text{if } y \text{ is the maximal suffix of } x0, \\ 0 & \text{otherwise.} \end{cases}$$

In general, the Markov chain method does not always lead to an efficient algorithm for computing the hitting probability of a spaced seed for the following two reasons. First, the number of states of the Markov chain grows exponentially fast with the number of zeros in Q . Indeed,

$$|\mathcal{S}| = (2 + r_0) + \sum_{k=1}^{L-w-1} (1 + r_k)2^k + r_{L-w}2^{L-w},$$

where we represent $Q = 1^{r_0} * 1^{r_1} \cdots * 1^{r_{L-w}}$. Here $r_0, r_{L-w} \geq 1$ and $r_i \geq 0, 1 \leq i < L - w$. Second, for some spaced seeds, there are also many cycles in the state diagrams. The number of cycles could grow exponentially fast with the number of zeros in Q for some spaced seeds. And hence, we need to introduce a system of recurrence relations.

Nevertheless, for some special spaced seeds, the corresponding Markov chain has a special structure where we can make use of to derive simple recurrence relations for the hitting probability. It is best illustrated by the following example and Proposition 2.2 below.

Example. Consider the spaced seed $1 * 1$, which is of weight 2 and length 3.

In this case, $\mathcal{S} = \{\varepsilon, 1, 11, 10, H\}$. The Markov chain for the spaced seed can be given by the state diagram in Fig. 1. Based on this Markov chain, we are able to derive a simple recurrence relation for \bar{Q}_n as

$$\bar{Q}_n = P[X_1, \dots, X_n \neq H | X_0 = \varepsilon].$$

For $n \geq 4$, from the state diagram, we obtain

$$\begin{aligned} \bar{Q}_n &= qP[X_2, \dots, X_n \neq H | X_1 = \varepsilon] + pP[X_2, \dots, X_n \neq H | X_1 = 1] \\ &= q\bar{Q}_{n-1} + pqP[X_3, \dots, X_n \neq H | X_2 = 10] + p^2P[X_3, \dots, X_n \neq H | X_2 = 11] \\ &= q\bar{Q}_{n-1} + pq^2P[X_4, \dots, X_n \neq H | X_3 = \varepsilon] + p^2qP[X_4, \dots, X_n \neq H | X_3 = 10] \\ &= q\bar{Q}_{n-1} + pq^2\bar{Q}_{n-3} + p^2q^2P[X_5, \dots, X_n \neq H | X_4 = \varepsilon] \\ &= q\bar{Q}_{n-1} + pq^2\bar{Q}_{n-3} + p^2q^2\bar{Q}_{n-4}. \end{aligned}$$

It is obvious that $\bar{Q}_n = 1, n = 0, 1, 2$ and $\bar{Q}_3 = 1 - p^2$.

And for consecutive seed of weight 2, we have $\bar{B}_0 = \bar{B}_1 = 1$ and

$$\bar{B}_n = q\bar{B}_{n-1} + pq\bar{B}_{n-2}, \quad n \geq 2.$$

Iterating the above equation once again, we have, for $n \geq 4$,

$$\bar{B}_n = q\bar{B}_{n-1} + pq^2\bar{B}_{n-3} + p^2q^2\bar{B}_{n-4}.$$

Therefore, for $D_n := \bar{Q}_n - \bar{B}_n, n \geq 0$,

$$D_n = qD_{n-1} + pq^2D_{n-3} + p^2q^2D_{n-4}, \quad n \geq 4, \tag{9}$$

and

$$D_0 = 0, \quad D_1 = 0, \quad D_2 = p^2, \quad D_3 = p^2 - p^3.$$

Hence, $D_n > 0$ for $n \geq 2$. In other words

$$B_{n-1} < Q_n < B_n, \quad n \geq 2.$$

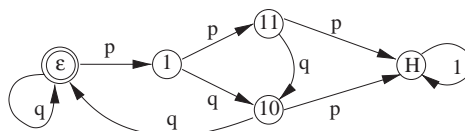


Fig. 1. The state diagram of the Markov chain for the space seed $1 * 1$.

The expectation of the first hitting time can be derived from the recurrence relation. We sum the recurrence relation for \bar{Q}_n for $n \geq 4$. After some algebraic simplification we arrive at

$$E[t_Q] = \frac{1 + 2p + p^2q}{p^2(1 + pq)}.$$

The example above can be further generalized to the next proposition with its proof given in Appendix A.

Proposition 2.2. *Let $Q = 1^a *^b 1$ be of length $L = a + b + 1$ and weight $a + 1$, where $0 < b \leq a$. Then, the hitting probability Q_n satisfies the recurrence relation,*

$$\bar{Q}_n = \sum_{j=1}^a p^{j-1} q \bar{Q}_{n-j} + \sum_{j=0}^{b-1} p^{a+j} q^{j+2} \bar{Q}_{n-L-j} + p^{a+b} q^{b+1} \bar{Q}_{n-L-b}, \quad n \geq L + b, \tag{10}$$

with initial values $\bar{Q}_n = 1$, $0 \leq n \leq L - 1$; and for $0 \leq j \leq b - 1$

$$\bar{Q}_{L+j} = 1 - (j + 1)p^{a+1} + p^{a+3} \frac{j - (j + 1)pq + (pq)^{j+1}}{(1 - pq)^2}.$$

3. Comparing spaced and consecutive seeds

Recall that A_i denotes the event that seed Q hits the random sequence $S = s_1 s_2 \dots s_i \dots$ at position i and \bar{A}_i the complement of A_i .

Theorem 3.1. *Let Q be a spaced seed of length L . Then, for any $2L - 1 \leq k \leq n$,*

- (a) $f_k \bar{Q}_{n-k+L-1} \leq f_n \leq f_k \bar{Q}_{n-k}$.
- (b) $\bar{Q}_k \bar{Q}_{n-k+L-1} \leq \bar{Q}_n < \bar{Q}_k \bar{Q}_{n-k}$.

Proof. Recall that $\bar{A}_{i,j} = \bar{A}_i \bar{A}_{i+1} \dots \bar{A}_j$ and $f_n = P[A_L \bar{A}_{L+1,n}]$ (see the proof of Proposition 2.1). The second inequality of (a) follows directly from the fact that $A_L \bar{A}_{L+1,n} \subseteq A_L \bar{A}_{L+1,k} \bar{A}_{k+L,n}$. Similarly, the second inequality follows directly from the fact that $\bar{A}_{L,n} \subset \bar{A}_{L,k} \bar{A}_{k+L,n}$.

We will prove the first inequality of (a) below. For $1 \leq i \leq L - 1$, let W_i be the set of all words of length i on the alphabet $\{0, 1\}$. For any $w \in W_i$, we use E_w to denote the event that $s_{k-L+2} s_{k-L+3} \dots s_{k-L+i+1} = w$. Recall ϵ denotes the empty word. We use E_ϵ to denote the sample space. Furthermore, it follows that $E_\epsilon = E_0 \cup E_1$. Obviously, for string w of length less than $L - 1$, $E_w = E_{w0} \cup E_{w1}$ and E_{w0} and E_{w1} are disjoint. By applying conditional probability, we have

$$\begin{aligned} f_n &= \sum_{w \in W_{L-1}} P[E_w] P[A_L \bar{A}_{L+1,k} \bar{A}_{k+1,n} | E_w] \\ &= \sum_{w \in W_{L-1}} P[E_w] P[A_L \bar{A}_{L+1,k} | E_w] P[\bar{A}_{k+1,n} | E_w] \end{aligned}$$

where the last equality follows from the fact that conditioned on E_w , with $w \in W_{L-1}$, the event $A_L \bar{A}_{L+1,k}$ is independent of the positions beyond position k and $\bar{A}_{k+1,n}$ is independent of the first $k - L + 1$ positions. On the other hand, $f_k = P[A_L \bar{A}_{L+1,k}] = P[A_L \bar{A}_{L+1,k} | E_c]$ and $\bar{Q}_{n-k+L-1} = P[\bar{A}_{k+1,n} | E_c]$. Thus, we only need to prove that

$$\begin{aligned} & \sum_{w \in W_j} P[E_w] P[A_L \bar{A}_{L+1,k} | E_w] P[\bar{A}_{k+1,n} | E_w] \\ & \geq \sum_{w \in W_{j-1}} P[E_w] P[A_L \bar{A}_{L+1,k} | E_w] P[\bar{A}_{k+1,n} | E_w] \end{aligned} \tag{11}$$

for any $1 \leq j \leq L - 1$ as follows.

For any $w \in W_{j-1}$, E_w is the disjoint union of E_{w0} and E_{w1} . By conditioning, we have

$$\begin{aligned} & P[A_L \bar{A}_{L+1,k} | E_w] P[\bar{A}_{k+1,n} | E_w] \\ & = (pP[A_L \bar{A}_{L+1,k} | E_{w1}] + qP[A_L \bar{A}_{L+1,k} | E_{w0}]) (pP[\bar{A}_{k+1,n} | E_{w1}] + qP[\bar{A}_{k+1,n} | E_{w0}]). \end{aligned} \tag{12}$$

Observe that $P[A_L \bar{A}_{L+1,k} | E_{w1}] \leq P[A_L \bar{A}_{L+1,k} | E_{w0}]$ and $P[\bar{A}_{k+1,n} | E_{w1}] \leq P[\bar{A}_{k+1,n} | E_{w0}]$. Hence, by applying Chebyshev's inequality to (12), we obtain that

$$\begin{aligned} & P[E_w] P[A_L \bar{A}_{L+1,k} | E_w] P[\bar{A}_{k+1,n} | E_w] \\ & \leq P[E_w] (pP[A_L \bar{A}_{L+1,k} | E_{w1}] P[\bar{A}_{k+1,n} | E_{w1}] + qP[A_L \bar{A}_{L+1,k} | E_{w0}] P[\bar{A}_{k+1,n} | E_{w0}]) \\ & = P[E_{w1}] P[A_L \bar{A}_{L+1,k} | E_{w1}] P[\bar{A}_{k+1,n} | E_{w1}] + P[E_{w0}] P[A_L \bar{A}_{L+1,k} | E_{w0}] P[\bar{A}_{k+1,n} | E_{w0}]. \end{aligned}$$

This implies inequality (11) since $W_j = \{w0, w1 \mid w \in W_{j-1}\}$. Hence, we finish the proof of the first inequality.

Since $\sum_{j=n+1}^{\infty} f_j = \bar{Q}_n$, the first inequality of (b) follows immediately as follows

$$\bar{Q}_k \bar{Q}_{n-k+L-1} = \sum_{j=k+1}^{\infty} f_j \bar{Q}_{n-k+L-1} \leq \sum_{j=k+1}^{\infty} f_{n-k+j} = \bar{Q}_n. \quad \square$$

From Theorem 3.1, we are able to derive tight lower and upper bounds on the expected value of the first time, t_Q , that the spaced seed Q hits the random sequence S . Since the hitting position is defined to be the ending position, the expected value $E(t_Q)$ here differs from the one defined by Keich et al. in [15] by a constant term L . Applying the first inequality of Theorem 3.1 (a) with $k = 2L - 1$, we have that

$$E(t_Q) = \sum_{n=0}^{\infty} P[t_Q > n] = L + \sum_{n=L}^{\infty} \bar{Q}_n \leq L + \sum_{j=2L}^{\infty} f_j / f_{2L-1} = L + \bar{Q}_{2L-1} / f_{2L-1}.$$

Similarly, applying the second inequality of Theorem 3.1 (a) with $k = 2L - 1$, we have that

$$E(t_Q) = L + \sum_{n=L}^{\infty} \bar{Q}_n \geq L + \sum_{j=3L-1}^{\infty} f_j / f_{2L-1} = L + \bar{Q}_{3L-2} / f_{2L-1}.$$

Therefore, we have proved the following theorem.

Theorem 3.2. Let Q be a spaced seed of length L and let t_Q be the first time Q hits an random sequence S . Then,

$$L + \bar{Q}_{3L-2}/f_{2L-1} \leq E(t_Q) \leq L + \bar{Q}_{2L-1}/f_{2L-1}.$$

Remark 1. These lower and upper bounds are quite tight since their difference is only

$$(\bar{Q}_{2L-1} - \bar{Q}_{3L-2})/f_{2L-1} = \sum_{i=2L}^{3L-2} f_i/f_{2L-1} \leq L - 1$$

following from the fact that $f_i \leq f_{2L-1}$ for each i in the range.

We now apply Theorem 3.1 to compare a spaced seed and the consecutive seed of the same weight. Let Q be the spaced seed of length L and weight w specified by its relative position set given in Formula (1). Assume $l = \text{gcd}(i_1, i_2, \dots, i_w) > 1$. Then, the following relative position set

$$\{i_1 = 0, i_2/l, i_3/l, \dots, i_w = (L - 1)/l\}$$

gives a new spaced seed Q' of the same weight w but length only L/l . For example, if $Q = 1 * 1 * * * 1 * 1$, then $Q' = 11 * 11$. Using Q'_n to denote the hitting probability of Q' on the random sequence S , we have

Proposition 3.1.

$$\bar{Q}_n = (\bar{Q}'_k)^{l-r} (\bar{Q}'_{k+1})^r > \bar{Q}'_n$$

where $k = \lfloor n/l \rfloor$ and $r = n - kl$.

Proof. The equality follows from the facts that Q hits a random sequence $S = s_1 s_2 \dots s_n$ if and only if Q' hits one of the following l random subsequences

$$s_1 s_{l+1} s_{2l+1} \dots s_{kl+1}, \dots, s_r s_{l+r} \dots s_{kl+r},$$

$$s_{r+1} s_{l+r+1} \dots s_{(k-1)l+r+1}, \dots, s_l s_{2l} \dots s_{kl}$$

and that Q' hits the first r random subsequences with probability Q'_{k+1} and the rest with Q'_k respectively.

The inequality follows from the second inequality of (b) in Theorem 3.1, which is actually true for any $k > 0$. \square

Applying Proposition 3.1 to the spaced seed $Q = 1 *^b 1$, $b \geq 1$, we obtain $Q' = 11$ and $\bar{Q}_n > \bar{Q}'_n$. Therefore, $Q_n < Q'_n$. (On the other hand, $Q'_{n-b} \leq Q_n$, by a lemma proved in [15].) We summarize this fact as

Proposition 3.2. Consecutive seed of weight 2 is optimal, i.e., most sensitive, among all seeds of weight 2.

In general, given a spaced seed Q of weight $w = w(Q)$, if its position set $\mathcal{RP}(Q)$ is equal to $\{0, 1, 2, \dots, (w - 1)\} \times l$ for some $l > 1$, then its hitting probability Q_n is smaller than the hitting probability of the consecutive seed with the same weight w . Such a class of spaced

patterns were used for approximate pattern matching in [19]. In addition, we have the following conjecture.

Conjecture. Let Q be a spaced seed of weight w and B the consecutive seed of the same weight. If the relative position set $\mathcal{RP}(Q) \neq \{0, 1, 2, \dots, w-1\} \times l$ for any integer $l > 1$, then, $Q_n \geq B_n$ for any $n > c$, where c is a constant depending on L and p .

Remark 2. (a) For any positive integer n , we define

$$\mu(n) := \max\{k \mid k \text{ is an integer such that } Q_n \geq B_{n+k}\}.$$

Obviously, $\mu(L) = -L(Q) + w(Q) < 0$. In fact, for any n , $\mu(n) \geq -L(Q) + w(Q)$ as proved in Keich et al. (2002). The conjecture above says that $\mu(n) \geq 0$ if n is large enough for the spaced seed satisfying the condition. Here we give a weak limiting result towards this conjecture. If $\mu(n_0) \geq w$ for some $n_0 > 0$, then, $\lim_{n \rightarrow \infty} \mu(n) = \infty$ (to be proved in Appendix C).

(b) Similarly, for any two spaced seeds Q, Q' , we define $\mu_{Q,Q'}(n)$ to be the largest integer k such that $Q_n \geq Q'_{n+k}$ for $n > 0$. If $\mu_{Q,Q'}(n_0) \geq |Q'|$, the length of Q' , then $\lim_{n \rightarrow \infty} \mu_{Q,Q'}(n) = \infty$.

4. Efficient algorithm for identifying the optimal seeds

4.1. Numerical analysis

In this subsection, we present numerical results in comparison of the spaced and consecutive seeds of the same weight. Our proposed screening method for identifying the optimal seeds is based on the theoretical results presented in Section 2 and the numerical analysis below.

Let Q be a spaced seed satisfying the relative position condition in the conjecture and B the consecutive seed of the same weight. We define, the *crossing position* of the spaced seed,

$$c(Q, p) = \min\{n \mid Q_n(p) \geq B_n(p)\}.$$

From our discussion in the last section, we know that $c(Q, p)$ does not exist for a spaced seed that does not satisfy the condition in the conjecture. Obviously, if it exists, its value depends on both the probability p and the seed itself. To understand how the probability value of p is related to $c(Q, p)$, we computed the $c(Q, p)$'s for different spaced seeds and p 's using formulas (4)–(6). We summarize them in Table 2 and Fig. 2. One of the spaced seeds appearing in Fig. 2 was the optimal seed reported in [18]. Among all spaced seeds of weight 11, this optimal seed has the largest hitting probability on a random 64-bit sequence when $p = 0.70$. Our computational results show that, among all the spaced seeds of weight 11, it consistently attains the smallest $c(Q, p)$ for p from 0.3 to 0.8. In addition, our computation shows that on a random 64-bit sequence, the consecutive seed has higher hitting probability when p is below 0.20. This fact is in contrast to our intuition! We also notice that $c(Q, p)$ is not a monotone function of p for some spaced seeds in which the ‘don’t care’ positions are not evenly distributed. The results in Table 1 indicates that $c(Q, p)$ takes a wide range of values for different spaced seeds. For example, $c(Q, 0.70) = 42$ for

Table 2

The top four spaced seeds among all the seeds of weight 7 and length 11, and their hitting probabilities on a 64-bit random sequence

	11**1*1*111	11*1**1*111	11*1*1**111	1*111**1*11
0.1	0.05398	0.05398	0.05398	0.05397
0.3	0.1140	0.1140	0.1139	0.1137
0.5	0.2937	0.2930	0.2929	0.2922
0.6	0.6622	0.6601	0.6599	0.6595
0.7	0.9365	0.9347	0.9346	0.9352

Table 3

The top four spaced seeds among all the seeds of weight 11 and length 18, and their hitting probabilities on a 64-bit random sequence

	Seed I	Seed II	Seed III	Seed IV
0.3	8.3×10^{-5}	8.3×10^{-5}	8.3×10^{-5}	8.3×10^{-5}
0.5	2.12×10^{-2}	2.12×10^{-2}	2.11×10^{-2}	2.11×10^{-2}
0.6	0.1317	0.1316	0.1314	0.1314
0.7	0.4671	0.4669	0.4661	0.4660

Seed I = 111 * 1 * * 1 * 1 * * 11 * 111, Seed II = 111 * * 1 * 11 * * 1 * 1 * 111, Seed III = 11 * 1 * 1 * 11 * * 1 * * 1111, and Seed IV = 111 * * 11 * 1 * * 1 * 1 * 111.

4.2. Algorithm

The above two observations suggest the following heuristic rules in reducing the search space of spaced seeds for finding optimal spaced seeds of weight w :

- (i) Consider only spaced seeds Q with the smallest crossing position for some p .
- (ii) The ‘don’t care’ positions of the good spaced seeds cannot be too clustered, that is, each block of consecutive ‘don’t care’ positions cannot exceed a small b_0 .

The implementation of the rule (i) is straightforward. The program considers only those spaced seeds with the smallest value. When the probability p is larger than 0.50, $\min\{c(Q, p) \mid w(Q) = w, l(Q) = L\}$ is empirically observed to be about $2L - w$ (see Table 1).

To search for the optimal spaced seed of weight w and length L , we set b_0 in the rule (ii) to be $\lceil (L - w)/(w - 1) + 2 \rceil$, which is slightly over the average number of ‘don’t care’ positions in each block. Let h be the proportion of the spaced seeds examined. The analysis in Appendix B indicates that h is smaller than 0.4 for L and w in the range of interest.

Recall that Q_n is the probability that a spaced seed Q hits the random sequence S before or at position n and f_n the probability that Q first hits S at position n . To speed up our algorithm, we further apply the following rule:

- (iii) Select the top 10 spaced seeds of length L based on the values of $Q_{2L} + (n - 2L)f_{2L}$. Then, compute the values of Q_{64} for these 10 seeds and choose the seed with the highest Q_{64} .

Table 4

The optimal seeds for different weights for searching a 64-bit random sequence

Weights	Optimal seeds	Weights	Optimal seeds
7	11**1*1*111	11	111*1**1*1**11*111
8	11**1**1*1*111	12	111*1*11*1**11*111
9	11*11*1*1**111	13	111*11*11**1*1*1111
10	11*11**11*1*111	14	111*111**1*11**1*1111

The spaced seed of weight 11 agrees with that found in [18].

It is easy to see that this rule speeds up 3 or 4 times for $L \leq 20$ since we only need to compute Q_n for $L \leq n \leq 2L$. The rationale behind this rule is that for medium size spaced seeds Q , $Q_{2L} + (n - 2L)f_{2L}$ approximates Q_{64} well. Numerically, for $n > 2L$, we have

$$Q_n - [Q_{2L} + (n - 2L)f_{2L}] = O((n - 2L)^2 p^{2w} (1 - p)^2)$$

since from taking $m = 2L$ in (8) in Proposition 2.1,

$$Q_n = Q_{2L} + (n - 2L)f_{2L} - \sum_{i=2L}^{n-1} (n - i)P[A_L \bar{A}_{L+1} \cdots \bar{A}_i A_{i+1}],$$

where p is the probability that 1 occurs at a position of S , w the weight of Q , and A_i the event that Q occurs at position i .

Our method turns out to be very efficient. All experiments were performed on a 500 MHz Pentium III PC with 900 Mbyte of memory. When it is used to identify the optimal spaced seed of weight 11 and length 18 for searching a 64 random bit sequence, our algorithm took only about 1 h 29 min, almost 10 times faster than the existing searching algorithms. Although our algorithm is heuristic, it consistently produces the optimal seed in each test case. In fact, our program outputs 10 good spaced seeds in each test; at least eight of them are among the top 10 best spaced seed. Using our program, we found all the optimal spaced seeds of weight from 7 to 14. These optimal spaced seeds for searching a 64-bit random sequence are listed in Table 4.

5. Conclusion

We have derived a set of recurrence relations for computing the sensitivity of a spaced seed and presented some theoretical results for comparing spaced and consecutive seeds. However, many questions on this topic have not been solved yet. Besides the conjecture posed in this paper, the following interesting problems are also open:

Question 1. Is the hitting probability $Q_n(p)$ of a spaced seed Q polynomial-time computable in terms of n , p , and the length L and weight w of Q ?

Question 2. How robust is an optimal spaced seed? More precisely, for fixed L and w , if Q is an optimal spaced seed among all spaced seeds of length L and weight w for a particular $0 < p < 1$ and for a large n' , will Q remain optimal for all p and for all $n \geq n'$?

Although our algorithm performs at least 10 times faster than the existing ones, it is still not fast enough for finding optimal spaced seeds of large weight (say $w = 20$) in a personal computer. As a future research topic, we will continue to improve our algorithm.

Acknowledgments

The authors would like to thank Ming Li and Bin Ma for useful discussions on the topic studied in the paper and for providing Ref. [15].

Appendix A. Proof of Proposition 2.2

Proof. Recall that $\bar{Q}_n = P[X_1, \dots, X_n \neq H | X_0 = \varepsilon]$. We will first establish the recurrence relation. Making use of the structure of the state diagram (cf. Fig. 3), we have

$$\begin{aligned} \bar{Q}_n &= \sum_{j=1}^a p^{j-1} q \bar{Q}_{n-j} + p^a P[X_{a+1}, \dots, X_n \neq H | X_a = 1^a] \\ &= \sum_{j=1}^a p^{j-1} q \bar{Q}_{n-j} + p^a \sum_{w \in \{0,1\}^b} P[w] P[X_{a+b+1}, \dots, X_n \neq H | X_{a+b} = 1^a w]. \end{aligned}$$

By partitioning $\{0, 1\}^b$ into $b + 1$ sets, namely, $C_j := \{1^j 0 v | v \in \{0, 1\}^{b-1-j}\}$ for $0 \leq j < b$; and $C_b = \{1^b\}$, we see that the second sum can be rewritten as

$$\begin{aligned} p^a \sum_{j=0}^{b-1} \sum_{1^j 0 v \in C_j} P[1^j 0 v] P[X_{a+b+1}, \dots, X_n \neq H | X_{a+b} = 1^{a+j} 0 v] \\ + p^{a+b} P[X_{a+b+1}, \dots, X_n \neq H | X_{a+b} = 1^{a+b}]. \end{aligned}$$

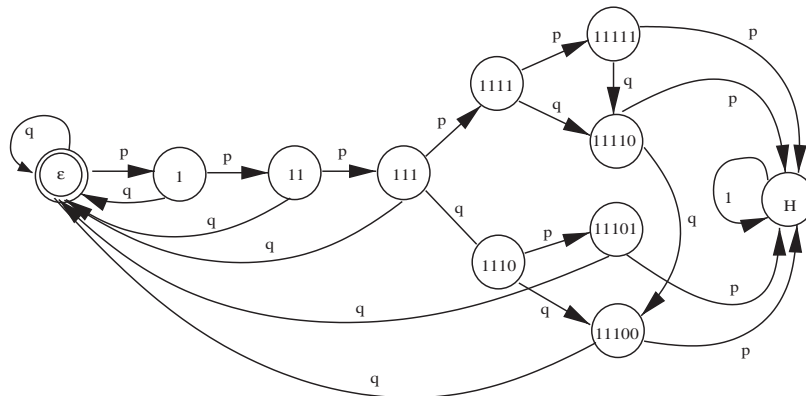


Fig. 3. The state diagram of the Markov chain for the space seed $1^a *^b 1$. Here $a = 3$ and $b = 2$.

To compute $P[X_{a+b+1}, \dots, X_n \neq H \mid X_{a+b} = 1^{a+b}]$, we proceed as follows. Now 1^{a+b} will either jump to H with probability p , or to $w' = 1^{a+b-1}0$. In order not to visit the state H up to the first n steps, there is only one such path, namely, $1^{a+b} \rightarrow 1^{a+b-1}0 \rightarrow 1^{a+b-2}0^2 \rightarrow \dots \rightarrow 1^a 0^b \rightarrow \varepsilon$ and start all over again. This happens with probability $q^{b+1} \bar{Q}_{n-L-b}$. This explains the last term in the recurrence relation (10).

Similarly, for $0 \leq j < b$. Let $w = 1^j 0 v$, there is again only one path, namely, $1^{a+j} 0 v \rightarrow 1^{a+j-1} 0 v 0 \rightarrow \dots \rightarrow 1^a 0 v 0^j \rightarrow \varepsilon$ and start all over again. This occurs with probability $q^{j+1} \bar{Q}_{n-L-j}$. Therefore,

$$p^a \sum_{v \in C_j} P[1^j 0 v] P[X_{a+b+1}, \dots, X_n \neq H \mid X_{a+b} = 1^{a+j} 0 v] = p^{a+j} q^{j+2} \bar{Q}_{n-L-j}.$$

Recurrence relation (10) follows immediately. As for the initial values, it is clear that $\bar{Q}_n = 1$ for $0 \leq n \leq L-1$, $\bar{Q}_L = 1 - p^{a+1}$ and $\bar{Q}_{L+1} = 1 - 2p^{a+1} + p^{a+3}$. If $2 \leq j \leq b-1$, then by (8) in Proposition 2.1,

$$\begin{aligned} \bar{Q}_{L+j} &= 1 - Q_{L+j} \\ &= 1 - (j+1)p^{a+1} + jp^{a+3} + \sum_{k=1}^{j-1} (j-k)p^{a+k+3} q^k \\ &= 1 - (j+1)p^{a+1} + \sum_{k=0}^j (j-k)p^{a+k+3} q^k \\ &= 1 - (j+1)p^{a+1} + p^{a+j+2} q^{j-1} \sum_{k=0}^j k(pq)^{-(k-1)} \\ &= 1 - (j+1)p^{a+1} + p^{a+3} [j - (j+1)pq + (pq)^{j+1}] / (1-pq)^2. \end{aligned}$$

We also used the fact that $P[A_L \bar{A}_{L+1, L+k} A_{L+k+1}] = p^{a+k+3} q^k$ for $1 \leq k \leq b-2$ in the second equality. \square

Appendix B. Analysis of the screening rule (ii)

Let h be the proportion of spaced seeds examined when rule (ii) is applied with $b_0 = \lceil (L-w)/(w-2) + 1 \rceil$. h can be shown equal to the probability of not finding $b_0 + 1$ consecutive ones in the probability space (equally likely) of all vectors of length $L-2$ with $w-2$ ones and $L-w$ zeros. Using the principle of inclusion and exclusion, we obtain

$$h = \sum_{i=0}^{w-2} (-1)^i \binom{w-1}{i} \binom{L-(b_0+1)i-2}{w-2} / \binom{L-2}{w-2}.$$

We use $|Q|$ to denote the length of a spaced seed Q . Since the optimal seed Q has about $0.7|Q|$ ones, we choose $w = \lfloor 0.7L \rfloor - \delta$, $0 \leq \delta \leq 3$ for our numerical analysis for $17 \leq L \leq 35$ (see Fig. 4). The analysis indicates that h is smaller than 0.4 for L in the range of interest.

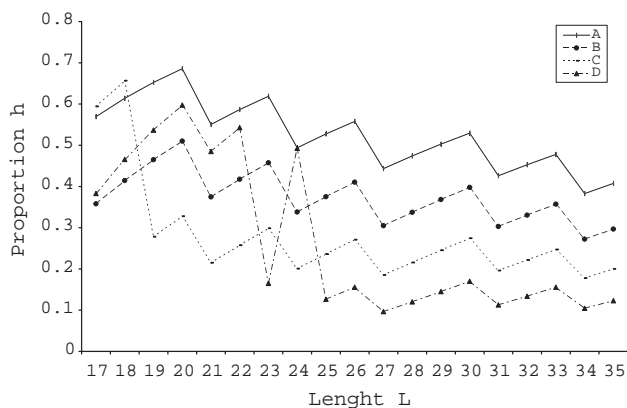


Fig. 4. Proportion of spaced seeds (h) examined for different L 's. Curves A–D correspond to $\delta = 0, 1, 2, 3$ in $w = \lfloor 0.7L \rfloor - \delta$.

Appendix C. Proof of Remark 2 in Section 3

Fact. If $\mu(n_0) \geq w$ for some $n_0 > 0$, then, $\lim_{n \rightarrow \infty} \mu(n) = \infty$.

Proof. By assumption, $Q_{n_0} \geq B_{n_0+w}$, or $\bar{Q}_{n_0} \leq \bar{B}_{n_0+w}$. By repeatedly applying the first inequality of (b) in Theorem 3.1 for the consecutive seed of weight w , we obtain

$$\begin{aligned} \bar{Q}_{kn_0} &\leq (\bar{Q}_{n_0})^k \leq (\bar{B}_{n_0+w})^k \\ &\leq \bar{B}_{2n_0+w+1} (\bar{B}_{n_0+w})^{k-2} \leq \dots \leq \bar{B}_{kn_0+w+k-1} \end{aligned}$$

for any $k \geq 1$. This implies $\mu(kn_0) \geq w + k - 1$. For any $n \geq 1$, there exists $k \geq 1$ such that $kn_0 \leq n < (k+1)n_0$, then

$$\bar{Q}_n \leq \bar{Q}_{kn_0} \leq \bar{B}_{kn_0+w+k-1} = \bar{B}_{n+(-n+kn_0+w+k-1)}$$

which, in turn, implies

$$\begin{aligned} \mu(n) &\geq -n + kn_0 + w + k - 1 \\ &> -(k+1)n_0 + kn_0 + w + k - 1 \\ &= w + k - 1 - n_0. \end{aligned}$$

As $n \rightarrow \infty$, $k \rightarrow \infty$ and hence $\mu(n) \rightarrow \infty$. \square

References

- [1] S.F. Altschul, et al., Basic local alignment search tool, J. Mol. Biol. 215 (1990) 403–410.
- [2] S.F. Altschul, et al., Gapped Blast and Psi-Blast: a new generation of protein database search programs, Nucleic Acids Res. 25 (1997) 3389–3402.

- [3] N. Balakrishnan, M.V. Koutras, *Runs and Scans with Applications*, Wiley, New York, USA, 2002.
- [4] B.E. Blaisdell, A measure of the similarity of sets of sequences not requiring sequence alignment, *Proc. Natl. Acad. Sci. USA* 83 (1986) 5155–5159.
- [5] J. Buhler, Efficient large-scale sequence comparison by locality-sensitive hashing, *Bioinformatics* 17 (2001) 419–428.
- [6] S. Burkhardt, J. Kärkkäinen, Better filtering with gapped q -grams, *Proceedings of the 12th Annual Symposium on Combinatorial Pattern Matching*, Jerusalem, Israel, 2001, pp. 73–85.
- [7] A. Califano, I. Rigoutsos, FLASH: fast look-up algorithm for string homology, Technical Report, IBM T. J. Watson Research Center, 1995.
- [8] A.L. Delcher, et al., Alignment of whole genomes, *Nucleic Acids Res.* 27 (1999) 2369–2376.
- [9] J.P. Dumas, J. Ninio, Efficient algorithms for folding and comparing nucleic acid sequences, *Nucleic Acids Res.* 10 (1982) 197–206.
- [10] J.C. Fu, M.V. Koutras, Distribution theory of runs: a Markov chain approach, *J. Amer. Statist. Assoc.* 89 (1994) 1050–1058.
- [11] W. Gish, WU-Blast 2.0, Website: <http://blast.wustl.edu>, 2001.
- [12] R.C. Hardison, J. Oeltjen, W. Miller, Long human-mouse sequence alignments reveal novel regulatory elements: a reason to sequence the mouse genome, *Genome Res.* 7 (1997) 966–969.
- [13] X. Huang, W. Miller, A time-efficient, linear-space local similarity algorithms, *Adv. Appl. Math.* 12 (1991) 337–357.
- [14] R. Karp, M.O. Rabin, Efficient randomized pattern-matching algorithms, *IBM J. Res. Develop.* 31 (1987) 249–260.
- [15] U. Keich, M. Li, B. Ma, J. Tromp, On spaced seeds, Manuscript, 2002.
- [16] W.-H. Li, Z. Gu, H. Wang, A. Nekrutenko, Evolutionary analysis of the human genome, *Nature* 409 (2001) 847–849.
- [17] D.J. Lipman, W.R. Pearson, Rapid and sensitive protein similarity searches, *Science* 227 (1985) 1435–1441.
- [18] B. Ma, J. Tromp, M. Li, PatternHunter-faster and more sensitive homology search, *Bioinformatics* 18 (2002) 440–445.
- [19] P. Pevzner, M.S. Watermann, Multiple filtration and approximate pattern matching, *Algorithmica* 13 (1995) 135–154.
- [20] F.P. Preparata, A. Frieze, E. Upfal, On the power of universal bases in sequencing by hybridization, *Proceedings of the third Annual International Conference on Computational and Molecular Biology*, Lyon, France, 1999, pp. 295–301.
- [21] F.P. Preparata, E. Upfal, Sequencing-by-hybridization at the information-theory bound: an optimal algorithm, *Proceedings of the 4th Annual International Conference on Computational and Molecular Biology*, Tokyo, Japan, 2000, pp. 245–253.
- [22] S.J. Schwager, Run probabilities in sequences of Markov-dependent trials, *J. Amer. Statist. Assoc.* 78 (1983) 168–175.
- [23] A.D. Solov'ev, A combinatorial identity and its application to the problem concerning the first occurrences of a rare event, *Theory Probab. Appl.* 11 (1966) 276–282.