

# Superiority and Complexity of the Spaced Seeds

Ming Li\*

Bin Ma<sup>†</sup>

Louxin Zhang<sup>‡</sup>

## Abstract

Optimal spaced seeds were introduced by the theoretical computer science community to bioinformatics to effectively increase homology search sensitivity. They are now serving thousands of homology search queries daily. While dozens of papers have been published on optimal spaced seeds since their invention, many fundamental questions still remain unanswered. In this paper, we settle several open questions in this area. Specifically, we prove that when the length of a non-uniformly spaced seed is bounded by an exponential function of the seed weight, the seed outperforms strictly the traditional consecutive seed in both (i) the average number of non-overlapping hits and (ii) the asymptotic hit probability. Then, we study the computation of the hit probability of a spaced seed, solving three more open questions: (iii) hit probability computation in a uniform homologous region is NP-hard and (iv) it admits a PTAS; (v) the asymptotic hit probability is computable in exponential time in seed length, independent of the homologous region length.

## 1 Introduction

Optimal spaced seeds are a theoretical computer science invention to increase the sensitivity and speed for homology search. They have been extensively studied recently. Homology search, or local alignment, finds similar segments between two DNA or protein sequences. It is the most fundamental and the most frequently performed task in bioinformatics. A large fraction of world's supercomputing time is currently consumed by homology search. The NCBI BLAST [1] server processes over  $10^5$  queries a day, which increase by 10-15% per month. By a different account, GenBank doubles in size every 18 months [24] which is at par with the growth rate of CPU speed. The inter-species comparative genomics research implies that homology search needs grow at a rate proportional to square of GenBank

size, quickly outgrowing the computer advances. Bigger and bigger clusters (over 1000 nodes) and parallel "BlastMachines" have been built to cope with this gigantic demand. Better algorithmic and mathematical solutions to this problem are thus indispensable.

In the 1970's, the dynamic programming technique [25, 30] was adopted to solve the problem "efficiently". It was quickly overwhelmed by the sea of biomolecular sequences.

In the 1980's, heuristics represented by FASTA [20] and BLAST [1] were introduced, trading sensitivity for speed. BLAST was designed based on the principle of filtration, where alignment between two sequences is found by first identifying short consecutive matches in specified positions, called *seed hits*, and then extending them for approximate matches, or *local alignments*. This approach faces a dilemma: setting the seed longer will cause many local alignments missing, resulting lower sensitivity; and setting the seed shorter generates too many random hits, resulting lower speed.

In PatterHunter [21], Ma, Tromp and Li introduced the idea of optimized spaced seeds to trigger a local alignment to increase both speed and sensitivity. More specifically, PatternHunter looks for runs of 18 consecutive nucleotide bases in each sequence, in which the nucleotide matches of a *hit* are only required at the eleven fixed positions specified by 1's in the string  $111*1**1*1**11*111$ , called *spaced seed*. It was noticed in [21] that such a spaced seed led to surprisingly higher sensitivity as well as speed. Moreover, further sensitivity improvement can be achieved by using multiple spaced seeds. PatternHunter was used by the Mouse Genome Sequence Consortium to compare the mouse and human genome sequences [14]. Recently, MegaBLAST, BLASTZ, next version of BLAST, and other alignment programs have also adopted the spaced seed approach.

For two spaced seeds of the same *weight*, i.e. same number of 1's, the expected number of hits is the same [21]. Intuitively, one might suspect that their sensitivities are also the same. However, the optimized spaced seed can improve sensitivity by as much as more than 50% [21]. The phenomenon is informally explained in terms of relaxing the correlations existing among consecutive sampling. However, rigorous theoretical

---

\*School of Computer Science, University of Waterloo, Waterloo, Ontario N2L 3G1, Canada, and City University of Hong Kong, Kowloon, Hong Kong, mli@uwaterloo.ca

<sup>†</sup>Department of Computer Science, University of Western Ontario, London, Ontario N6A 5B8, Canada. bma@csd.uwo.ca

<sup>‡</sup>Department of Mathematics, National University of Singapore, Singapore 117543. matzlx@nus.edu.sg.

analysis of this behavior is extremely hard. So far, it is only known that (i) with the same weight, the consecutive seed has higher hit probability than so called *uniformly spaced seeds*  $(1*^k)^m 1$ , where  $k, m \geq 1$  [7, 6] in any homologous region and hence not all spaced seeds are better; and (ii) non-uniformly spaced seeds may outperform consecutive seeds under various conditions, [16, 6, 7], but there have been no strict and definite statements regarding their relative power. To elucidate the mechanism that confers power to spaced seeds, Buhler, Keich and Sun [6] proposed an asymptotic modeling of the problem; Preparata, Zhang and Choi [27] proposed a probability leakage model.

This paper is to formally answer the above open problems. Since the overlapping hits can only be extended into one local alignment between the two sequences, the sensitivity of a spaced seed depends largely on the expected number of non-overlapping hits. We prove that, for a *non-uniformly* spaced seed  $Q$  of weight  $w_Q$  and length  $L_Q$ , the expected number of its non-overlapping hits,  $\mu_Q$ , in a Bernoulli sequence generated with probability  $p$  is strictly larger than the consecutive seed  $B$  of the same weight when  $L_Q < w_Q + \frac{1-p}{p}((\frac{1}{p})^{w_Q-2} - 1)$ . This explains clearly that non-uniformly spaced seeds  $Q$  are more sensitive than  $B$ . We note that when  $p < 1$  is fixed, the bound is an exponential function of the seed weight.

Another indicator of sensitivity is the hit probability  $Q_n(p)$  for a spaced seed  $Q$ , in a Bernoulli sequence of length  $n$  generated with probability  $p$ . The study of the hit probability of single or multiple patterns can be traced back to run statistics and renewal theory [3, 9, 31, 15, 28]. The general theory focuses on exact and asymptotic distribution of pattern occurrences, while the theory developed here is about seed comparison in terms of hitting probability. Based on a general theorem of Nicodéme *et al.* [26], Buhler *et al.* proved that there are two constants  $\alpha_Q$  and  $\lambda_Q$  determined only by  $Q$  and  $p$ , such that  $\lim_{n \rightarrow \infty} (1 - Q_n(p)) / (\alpha_Q \lambda_Q^n) = 1$  ([6]; see also [31]). It was then conjectured that  $\lambda_Q$  of a non-uniformly spaced seed  $Q$  is smaller than that of the consecutive seed  $B$  with the same weight [6]. In this paper, by providing tight lower and upper bounds on  $\lambda_Q$  in terms of  $\mu_Q$ , we prove that the conjecture is true when  $L_Q < \frac{1-p}{p}((\frac{1}{p})^{w_Q-2} - 1) + 1$ . Therefore, if  $L_Q < \frac{1-p}{p}((\frac{1}{p})^{w_Q-2} - 1) + 1$ , then  $Q_n(p) > B_n(p)$  for large  $n$ . This also answers the conjecture posed in the section 3 of [7].

Another important and practical problem is to find the most sensitive seed. A direct approach to finding the most sensitive seed is through exhaustive search after the hit probability of each spaced seed is calculated. The hit probability of a spaced seed can be computed by

dynamic programming [16] (see also [5, 6, 17, 19, 32] for various generalizations) or a recurrent relation [7]. This approach quickly becomes impractical because (i) the number of spaced seeds of length  $L$  and weight  $w$  grows exponentially in  $L - w$ ; (ii) the time complexity of the dynamic programming algorithm or recurrent relation based method is polynomial in the homologous region length  $n$  but exponential in  $L - w$ . It remains a major open problem whether computing the hit probability of a single spaced seed in a uniform region is NP-hard [21, 19]. With much effort, it was proved in [19] that computing the hit probability of a seed in non-uniform regions, or of multiple seeds in uniform regions, are NP-hard. It was also recently brought to our attention that [10] proved the determination of whether a seed can hit all regions with fixed length  $L$  and  $m$  mismatches is NP-hard.<sup>1</sup> However, the original open problem remains unsolved. In this paper, we show that computing the hit probability in a uniform region is indeed NP-hard, via a sophisticated counting argument. We then give an algorithm that computes the asymptotic sensitivity of a spaced seed in time independent of region length, by extending an eigenvalue argument of [6, 26]. This provides an algorithm to effectively compare the asymptotic sensitivity of two seeds. A polynomial time algorithm that approximately computes the hit probability with any fixed small error ratio is also provided.

## 2 Notations and preliminaries

Since the publication of [21], the spaced seed problem has been extensively studied in the Bernoulli sequence model [8, 12, 34, 16, 17, 19, 32] and more general Markov and HMM models [5, 6]. For simplicity, we restrict ourselves to the Bernoulli or zero-th order Markov sequence model in this paper, although most of our results generalize to higher order Markov models. Following the original PatternHunter paper [21], a non-gapped alignment of two DNA sequences  $S'$  and  $S''$  of length  $n$  corresponds a 0-1 sequence  $S$  of length  $n$ : 0 means a mismatch and 1 a match.  $S$  is modeled as a 0-1 Bernoulli random sequence of length  $n$  in which 1 is generated with probability  $p$ .

Let  $R$  be an infinite Bernoulli random sequence. We use  $R[k]$  to denote the  $k$ th symbol of  $R$  and  $R[0, k - 1]$  the length- $k$  prefix of  $R$  for  $k \geq 0$ . A *spaced seed*  $Q = q_0 q_1 \cdots q_{L_Q-1}$  is represented by a string over alphabet  $\{1, *\}$ .  $L_Q$  is called the *length* of  $Q$ . And the number of 1s in  $Q$ , denoted by  $w_Q$ , is called the *weight*

<sup>1</sup>This does not imply the NP-hardness for uniform regions because a uniform region with similarity  $m/L$  may contain more than  $m$  mismatch.

of  $Q$ . For the purpose of homology search, we always require  $q_0 = q_{L_Q-1} = 1$ .

The ‘1’-positions of a spaced seed  $Q$  define the following *relative position set*:

$$(2.1) \quad \mathcal{RP}(Q) = \{i_1 = 0, i_2, \dots, i_{w_Q} = L_Q - 1\}.$$

Thus, the seed  $Q$  is said to hit  $R$  *ending* at position  $k$  if and only if  $R[k - L_Q + i_j + 1] = 1$  for all  $1 \leq j \leq w_Q$ . Correspondingly,  $Q$  is said to hit  $R$  *starting* at position  $k$  if and only if  $R[k + i_j + 1] = 1$ . In this paper, if not explicitly specified, a hit at position  $k$  means the hit ends at position  $k$ . This is to follow the convention in renewal theory [9].

Let  $A_j$  be the event that  $Q$  hits  $R$  at position  $j$  and  $\bar{A}_j$  be the complement of  $A_j$ . Let  $f_j$  be the probability that  $Q$  **first** hits  $R$  at position  $j-1$ . That is,  $f_j = P[\bar{A}_0 \bar{A}_1 \cdots \bar{A}_{j-2} A_{j-1}]$ . Let  $Q_n := Q_n(p)$  denote the probability that  $Q$  hits  $R[0, n-1]$  and  $\bar{Q}_n := 1 - Q_n$ . The *sensitivity* of a spaced seed  $Q$  over the above length- $n$  homologous region  $R[0, n-1]$  is thus defined to be  $Q_n(p)$  given the *similarity level*  $p$ .

When  $0 \leq n \leq L_Q - 2$ , obviously  $A_n = \emptyset$  and  $Q_n = 0$ . For general  $n$ , we have

$$(2.2) \quad \bar{Q}_n = P[\cap_{j \leq n-1} \bar{A}_j] = \bar{Q}_{n-1} - f_n = \sum_{i > n} f_i.$$

### 3 Distance between non-overlapping hits

Renewal theory studies certain recurrent events connected with repeated trials. Roughly speaking, an event  $\mathcal{E}$  qualifies for the theory if after each occurrence of  $\mathcal{E}$ , the trials start from scratch [9]. Therefore, the number of trials between successive occurrences of  $\mathcal{E}$  are jointly independent random variables having the identical distribution. It is easy to see that a non-overlapping hit of a spaced seed  $Q$  is a recurrent event with the following convention: If a hit at position  $i$  is selected as a *non-overlapping hit*, then the next non-overlapping hit is the first hit at or after position  $i + L_Q$ .

This section focuses on the average distance,  $\mu_Q$ , between two successive non-overlapping hits of a spaced seed. We first give a formula for computing  $\mu_Q$  using technique in [13]; then, we give a simple upper bound on  $\mu_Q$ . This simple bound will be used in the next section.

**3.1 A formula for computing  $\mu_Q$ .** Let  $\mathcal{W}_Q$  be the set of all  $m := 2^{L_Q - w_Q}$  distinct strings obtained from the seed  $Q$  by filling 0 or 1 in the ‘‘don’t care’’ positions. For example, if  $Q = 1*11*1$ , we have  $\mathcal{W}_Q = \{101101, 101111, 111101, 111111\}$ . The seed  $Q$  hits at position  $n$  if and only if there is a  $W_j \in \mathcal{W}_Q$  occurring at the position. For each  $j$ , we use  $A_n^{(j)}$  to denote the event that the pattern  $W_j$  occurs at the position  $n$ . Clearly,

$A_n = \cup_{1 \leq j \leq m} A_n^{(j)}$ , and  $A_n^{(j)}$ ’s are disjoint. Setting

$$f_n^{(j)} = P[\bar{A}_0 \bar{A}_1 \cdots \bar{A}_{n-2} A_{n-1}^{(j)}], \quad 1 \leq j \leq m,$$

we have  $f_n = \sum_{1 \leq j \leq m} f_n^{(j)}$  and so Formula (2.2) becomes

$$(3.3) \quad \bar{Q}_n = \bar{Q}_{n-1} - f_n^{(1)} - f_n^{(2)} - \cdots - f_n^{(m)}.$$

For any  $W_j \in \mathcal{W}_Q$  and  $0 \leq a < b \leq L_Q - 1$ , we use  $W_j[a, b]$  to denote the substring of  $W_j$  from position  $a$  to position  $b$  inclusively. For any  $i, j$  and  $k$ ,  $1 \leq i, j \leq m$ ,  $1 \leq k \leq L_Q$ , define

$$p_k^{(ij)} = \begin{cases} v_{jk} & \text{if } W_i[L_Q - k, L_Q - 1] = W_j[0, k - 1] \\ 1 & k = L_Q \text{ \& } i = j \\ 0 & \text{otherwise} \end{cases}$$

where  $v_{jk}$  is the probability that  $W_j[k, L_Q - 1]$  hits at a position  $n \geq L_Q - 1$ . In other words,  $p_k^{(ij)}$  is the conditional probability that  $W_j$  occurs at the position  $a + (L_Q - k)$  given that  $W_i$  occurs at a position  $a$ .

LEMMA 3.1. ([7]). *Let  $p_j$  ( $1 \leq j \leq m$ ) be the probability that the pattern  $W_j \in \mathcal{W}_Q$  hits at position  $L_Q - 1$ . Then, for any  $1 \leq j \leq m$ ,*

$$(3.4) \quad \bar{Q}_n p_j = \sum_{i=1}^m \sum_{k=1}^{L_Q} f_{n+k}^{(i)} p_k^{(ij)}$$

To find the average distance  $\mu_Q$  between non-overlapping hits of  $Q$ , we define the generating functions

$$U(x) = \sum_{n=0}^{\infty} \bar{Q}_n x^n, \quad F_i(x) = \sum_{n=0}^{\infty} f_n^{(i)} x^n, \quad i \leq m.$$

By definition,  $\mu_Q = \sum_{j \geq L_Q} j f_j$ . Applying Formula (2.2), we have

$$\mu_Q = L_Q + \sum_{j \geq L_Q} \bar{Q}_j = U(1)$$

and both  $U(x)$  and  $F_i(x)$  converge when  $x \in [0, 1]$ . Multiplying (3.3) by  $x^{n-1}$  and summing on  $n$ , we obtain

$$(1 - x)U(x) + F_1(x) + F_2(x) + \cdots + F_m(x) = 1.$$

Similarly, by (3.4), we obtain

$$-x^{L_Q} p_j U(x) + \sum_{1 \leq i \leq m} C_{ij}(x) F_i(x) = 0, \quad 1 \leq j \leq m$$

where  $C_{ij}(x) = \sum_{k=1}^{L_Q} p_k^{(ij)} x^{L_Q - k}$ . Solving the above linear functional equations, and setting  $x = 1$ , we obtain the following formula for computing  $\mu_Q$ .

**THEOREM 3.1.** Let  $A_Q = [C_{ij}(1)]_{m \times m}$  and  $M_Q = \begin{bmatrix} 0 & I_{1 \times m} \\ P & A_Q \end{bmatrix}$ , where  $I = [1]_{1 \times m}$  and  $P = [-p_i]_{m \times 1}$ . Then,  $\mu_Q = \det(A_Q)/\det(M_Q)$ .

**Remarks** (a) Using the above theorem, one can easily show that  $\mu_B = \sum_{i=1}^w p^{-i}$ .

(b) For spaced seed  $Q = 1^a * 1^b$ ,  $a \geq b \geq 1$ , then,  $\mathcal{W}_Q = \{W_1, W_2\} = \{1^a 0 1^b, 1^{a+b+1}\}$  and

$$A_Q = \begin{bmatrix} \sum_{i=0}^{b-1} p^{a+i} q + 1 & \sum_{i=0}^{a-1} p^{b+i} q \\ \sum_{i=0}^{b-1} p^{a+1+i} & \sum_{i=0}^{a+b} p^i \end{bmatrix}.$$

$$\text{Hence, } \mu_Q = \frac{\sum_{i=0}^{a+b} p^i + \sum_{i=0}^b \sum_{j=0}^{b-1} p^{a+i+j} q}{p^{a+b}(1 + \sum_{i=1}^b p^i q)}.$$

**3.2 An upper bound for  $\mu_Q$ .** A spaced seed  $Q$  is *uniform* if its relative match positions form an arithmetic sequence. For example,  $1^{**1**1}$  is uniform with relative matching position set  $\{0, 3, 6\}$  in which the difference between two successive positions is 3. Any spaced seed of weight 2 is uniform. By definition, we have that  $\mu_Q \geq L_Q$ . Thus, for any fixed  $p$ , the expected distance  $\mu_Q$  between two successive non-overlapping hits will be larger than that of the consecutive seed of the same weight when  $L_Q$  is large enough. However, we can show that when  $L_Q$  is not too long,  $\mu_Q$  is always smaller than that of the consecutive seed.

For any  $0 \leq j \leq L_Q - 1$ , define  $\mathcal{RP}(Q) + j = \{l_1 + j, l_2 + j, \dots, l_{w_Q} + j\}$  and let  $o_Q(j) = |\mathcal{RP}(Q) \cap (\mathcal{RP}(Q) + j)|$ . In other words,  $o_Q(j)$  is the number of 1's that **overlap** between the seed and the  $j$ -th shifted version of it. Trivially,  $o_Q(0) = w_Q$  and  $o_Q(L_Q - 1) = 1$  for any seed  $Q$ .

**THEOREM 3.2.** For any non-uniformly spaced seed  $Q$ ,

$$\mu_Q \leq \sum_{i=1}^{w_Q} p^{-i} + (L_Q - w_Q) - \frac{q}{p} (p^{-(w_Q-2)} - 1)$$

*Proof.* Using martingale theory, Keich *et al.* [16] proved that  $\mu_Q \leq \sum_{i=0}^{L_Q-1} p^{-o_Q(i)}$ . So, we only need to prove that

$$\leq \sum_{i=0}^{L_Q-1} p^{-o_Q(i)} \leq \sum_{i=1}^{w_Q} p^{-i} + (L_Q - w_Q) - \frac{q}{p} (p^{-(w_Q-2)} - 1)$$

for any non-uniformly spaced seed  $Q$  by induction on the weight  $w_Q$ . Obviously, this is true for  $w_Q = 3$ .

Assume  $\mathcal{RP}(Q) = \{l_1 = 0, l_2, \dots, l_{w_Q} = L_Q - 1\}$  and let  $Q = 1^{*r} Q'$ , where  $r = l_2 - 1 \geq 0$ . Now, we consider  $Q'$  as a spaced seed. Obviously,  $L_{Q'} = L_Q - r - 1$  and  $w_{Q'} = w_Q - 1$ . Assume there are  $k$  letters

'1' in  $Q[0, L_Q - r - 2]$ . Then, there are  $w_Q - k$  letters '1' and  $r + 1 + k - w_Q$  letters '\*' in the  $Q[L_Q - r - 1, L_Q - 1]$ . Hence, we have, for  $0 \leq i \leq L_Q - r - 2$ ,

$$(3.5) \quad o_Q(i) = \begin{cases} o_{Q'}(i) + 1 & i = l_1, l_2, \dots, l_k \\ o_{Q'}(i) & \text{otherwise} \end{cases}$$

and for  $L_Q - r - 1 \leq i \leq L_Q - 1$ ,

$$(3.6) \quad o_Q(i) = \begin{cases} 1 & i = l_{k+1}, l_2, \dots, l_w \\ 0 & \text{otherwise} \end{cases}$$

Since  $l_1 = 0$ , we also have  $o_Q(l_1) = w_Q = 1 + w_{Q'} = 1 + o_{Q'}(l_1)$ . Using (3.5) and (3.6) and  $\frac{1}{p} = 1 + \frac{q}{p}$ , we first have

$$\begin{aligned} & \sum_{i=0}^{L_Q-1} \left(\frac{1}{p}\right)^{o_Q(i)} \\ &= \sum_{i=0}^{L_Q-r-2} \left(\frac{1}{p}\right)^{o_Q(i)} + \sum_{i=L_Q-r-1}^{L_Q-1} \left(\frac{1}{p}\right)^{o_Q(i)} \\ &= \left[ \sum_{j=1}^k \left(\frac{1}{p}\right)^{o_{Q'}(l_j)+1} + \sum_{i=0, i \notin \mathcal{RS}(Q)}^{L_Q-r-2} \left(\frac{1}{p}\right)^{o_{Q'}(i)} \right] \\ & \quad + \left[ \sum_{j=k+1}^{w_Q} \frac{1}{p} + \sum_{i=L_Q-r-1, i \notin \mathcal{RS}(Q)}^{L_Q-1} \left(\frac{1}{p}\right)^0 \right] \\ &= \left[ \sum_{j=1}^k \left(\frac{1}{p}\right)^{o_{Q'}(l_j)+1} + \sum_{i=0, i \notin \mathcal{RS}(Q)}^{L_Q-r-2} \left(\frac{1}{p}\right)^{o_{Q'}(i)} \right] \\ & \quad + [(w_Q - k) \left(\frac{1}{p}\right)^1 + r + 1 + (k - w_Q)] \\ &= \left[ \sum_{j=1}^k \left(\frac{1}{p}\right)^{o_{Q'}(l_j)+1} + \sum_{i=0, i \notin \mathcal{RS}(Q)}^{L_Q-r-2} \left(\frac{1}{p}\right)^{o_{Q'}(i)} \right] \\ & \quad + (w_Q - k) \frac{q}{p} + r + 1 \\ &= \left[ \left(1 + \frac{q}{p}\right) \sum_{j=1}^k \left(\frac{1}{p}\right)^{o_{Q'}(l_j)} + \sum_{i=0, i \notin \mathcal{RS}(Q)}^{L_Q-r-2} \left(\frac{1}{p}\right)^{o_{Q'}(i)} \right] \\ & \quad + (w_Q - k) \frac{q}{p} + r + 1 \\ &= \sum_{i=0}^{L_Q-r-2} \left(\frac{1}{p}\right)^{o_{Q'}(i)} + \frac{q}{p} \sum_{j=1}^k \left(\frac{1}{p}\right)^{o_{Q'}(l_j)} \\ & \quad + (w_Q - k) \frac{q}{p} + r + 1 \end{aligned}$$

Replacing  $x$  by  $\frac{1}{p}$  in the formula

$x^{w_Q} - 1 = (x - 1) \sum_{i=0}^{w_Q-1} x^i$ , we obtain  $\left(\frac{1}{p}\right)^{w_Q} = 1 + \frac{q}{p} \sum_{i=0}^{w_Q-1} \left(\frac{1}{p}\right)^i$ . Replacing 1 by  $\left(\frac{1}{p}\right)^{w_Q} - \frac{q}{p} \sum_{i=0}^{w_Q-1} \left(\frac{1}{p}\right)^i$  and grouping the terms having  $\frac{q}{p}$  together, we further have

$$\begin{aligned} & \sum_{i=0}^{L_Q-1} \left(\frac{1}{p}\right)^{o_Q(i)} \\ &= \left(\frac{1}{p}\right)^{w_Q} + \sum_{i=0}^{L_Q-r-2} \left(\frac{1}{p}\right)^{o_{Q'}(i)} + r \\ & \quad + \frac{q}{p} \left[ \sum_{j=1}^k \left(\frac{1}{p}\right)^{o_{Q'}(l_j)} + (w_Q - k) - \sum_{i=0}^{w_Q-1} \left(\frac{1}{p}\right)^i \right] \\ &= \left(\frac{1}{p}\right)^{w_Q} + \sum_{i=0}^{L_Q-r-1} \left(\frac{1}{p}\right)^{o_{Q'}(i)} + r \\ & \quad + \frac{q}{p} \left[ \sum_{j=2}^k \left(\frac{1}{p}\right)^{o_{Q'}(l_j)} + (w_Q - k) - \sum_{i=0}^{w_Q-2} \left(\frac{1}{p}\right)^i \right]. \end{aligned}$$

Now, we consider the following two cases.

*Case 1.* The seed  $Q'$  is uniform. Assume the matching positions of  $Q'$  form an arithmetic sequence with difference  $s$ . Since  $Q$  is non-uniform,  $r \neq s$ , and hence,  $o_{Q'}(l_j) \leq w_Q - j - 1$  for  $j = 2, 3, \dots, k$ . Therefore,

$$\sum_{j=2}^k \left(\frac{1}{p}\right)^{o_{Q'}(l_j)} + (w_Q - k) - \sum_{i=0}^{w_Q-2} \left(\frac{1}{p}\right)^i \leq -\left(\frac{1}{p}\right)^{(w_Q-2)} + 1.$$

Since  $Q'$  is uniform,

$$\begin{aligned} \sum_{i=0}^{L_{Q'}-1} \left(\frac{1}{p}\right)^{o_{Q'}(i)} &= \sum_{i=1}^{w_{Q'}} \left(\frac{1}{p}\right)^i + L_{Q'} - w_{Q'} \\ &= \sum_{i=1}^{w_Q-1} \left(\frac{1}{p}\right)^i + L_Q - w_Q - r. \end{aligned}$$

Hence,

$$\sum_{i=0}^{L_Q-1} \left(\frac{1}{p}\right)^{o_Q(i)} \leq \sum_{i=1}^{w_Q} \left(\frac{1}{p}\right)^i + (L_Q - w_Q) - \frac{q}{p} \left[ \left(\frac{1}{p}\right)^{w_Q-2} - 1 \right].$$

*Case 2.* The seed  $Q'$  is not uniform. By induction,

$$\begin{aligned} & \sum_{i=0}^{L_{Q'}-1} \left(\frac{1}{p}\right)^{o_{Q'}(i)} \\ & \leq \sum_{i=1}^{w_{Q'}} \left(\frac{1}{p}\right)^i + L_{Q'} - w_{Q'} - \frac{q}{p} \left[ \left(\frac{1}{p}\right)^{w_{Q'}-2} - 1 \right] \\ & = \sum_{i=1}^{w_Q-1} \left(\frac{1}{p}\right)^i + L_Q - w_Q - r - \frac{q}{p} \left[ \left(\frac{1}{p}\right)^{w_Q-3} - 1 \right]. \end{aligned}$$

Since  $Q'$  is not uniform,  $o_{Q'}(l_2) = o_Q(l_2) - 1 \leq w_Q - 2 - 1 = w_Q - 3$  and  $o_{Q'}(l_j) \leq w_Q - j$ ,  $j \geq 3$ . Hence,

$$\begin{aligned} & \sum_{i=0}^{L_Q-1} \left(\frac{1}{p}\right)^{o_Q(i)} \\ & \leq \sum_{i=1}^{w_Q} \left(\frac{1}{p}\right)^i + (L_Q - w_Q) + \frac{q}{p} \left[ 1 - \left(\frac{1}{p}\right)^{w_Q-3} \right] \\ & \quad + \sum_{j=2}^k \left(\frac{1}{p}\right)^{o_{Q'}(l_j)} + (w_Q - k) - \sum_{i=0}^{w-2} \left(\frac{1}{p}\right)^i \\ & \leq \sum_{i=1}^{w_Q} \left(\frac{1}{p}\right)^i + (L_Q - w_Q) - \frac{q}{p} \left[ \left(\frac{1}{p}\right)^{w_Q-2} - 1 \right]. \end{aligned}$$

#### 4 The average number of non-overlapping hits

Renewal theory shows that the number of the non-overlapping hits of a spaced seed  $Q$  in a long Bernoulli random sequence of length  $N$  has, approximately, a normal distribution with mean  $\frac{N}{\mu_Q}$ . Recall, for the consecutive seed  $B$  of weight  $w$ ,  $\mu_B = \sum_{i=1}^w \left(\frac{1}{p}\right)^i$ , by Theorem 3.2, we have

**THEOREM 4.1.** *Given any non-uniformly spaced seed  $Q$  of length  $L_Q$  and weight  $w_Q$ . If  $L_Q < w_Q + \frac{q}{p} \left[ \left(\frac{1}{p}\right)^{w_Q-2} - 1 \right]$ , then,  $Q$  has on average more non-overlapping hits than the consecutive seed of the same weight in homologous regions with sequence similarity  $p$  in the Bernoulli model.*

This explains clearly why a homology search program with a good spaced seed is much more sensitive than with the consecutive seed of the same weight although it generates on average fewer number of hits. In homology search, the weight of the seed spaced used is about 11 and sequence similarity is roughly 70%. When a spaced seed  $Q$  of weight  $w_Q$  and length  $L_Q < 5w_Q/2$  is used, by Theorem 3.2, the program will identify on average  $\frac{N}{\mu_Q}$  non-overlapping hits in a homologous region of length  $N$ , which is much larger than the corresponding number when the consecutive seed of the same weight is used. Since overlapping hits can only be extended into one local alignment, the program with a spaced seed generates more local alignments and hence is more sensitive.

#### 5 Asymptotic analysis of the hit probability

Because of its larger span, in terms of the hit probability  $Q_n$ , a spaced seed  $Q$  usually lags behind the consecutive

seed  $B$  of the same weight for small  $n$  and then surpasses  $B$  when  $n$  is large enough. To compare spaced seed efficiently, Buhler *et al.* proposed the asymptotic analysis of spaced seeds. In [6], they proved that for any spaced seed  $Q$ , there are two constants  $\alpha_Q$  and  $\lambda_Q$  that do not depend on  $n$  such that  $\lim_{n \rightarrow \infty} \bar{Q}_n / (\alpha_Q \lambda_Q^n) = 1$  (see also [26]), where  $\lambda_Q$  is the largest eigenvalue of some transition matrix of a Markov Chain model constructed for computing the hit probability  $Q_n$ . Surprisingly, Solov'ev proved a similar result in a more general setting four decades ago [31].

To compare spaced seeds, we first establish tight lower and upper bounds on  $\lambda_Q$  using  $\mu_Q$  studied in Section 3.

**THEOREM 5.1.** (1) *For the consecutive seed  $B$  of weight  $w$ ,  $\lambda_B$  satisfies the following inequalities:  $1 - \frac{1}{\sum_{i=1}^w (1/p)^{i-w+1}} \leq \lambda_B \leq 1 - \frac{1}{\sum_{i=1}^w (1/p)^{i-w} + \sum_{i=0}^{w-1} p^i}$ .*

(2) *For a spaced seed  $Q$ ,*  
 $1 - \frac{1}{\mu_Q - L_Q + 1} \leq \lambda_Q \leq 1 - \frac{1}{\mu_Q}$ .

*Proof.* (1). Noticing that  $\mu_B = \sum_{i=1}^w p^{-i}$ , we can derive the first inequality from the corresponding one in (2). Now, we prove the second one.

Since  $B$  is consecutive and of weight  $w$ , the event that  $B$  hits  $R$  at position  $n+1$  but not at  $n$  occurs if and only if  $R[n-w+1, n+1] = 011\dots 1$ . Let  $A_i$  denote the event that the seed  $B$  occurs at position  $i$  and  $\bar{A}_i$  the complement of  $A_i$ . Then, we have  $b_{n+1} := P[\bar{A}_0 \bar{A}_1 \dots \bar{A}_{n-1} A_n] = p^w q \bar{B}_{n-w}$ , where  $q = 1 - p$ , and therefore

$$\begin{aligned} & \bar{B}_n \\ & = P[\bar{A}_0 \bar{A}_1 \dots \bar{A}_{n-2} \bar{A}_{n-1}] \\ & = P[\bar{A}_0 \bar{A}_1 \dots \bar{A}_{n-w-1} \bar{A}_{n-1}] \\ & \quad - \sum_{j=0}^{w-2} P[\bar{A}_0 \bar{A}_1 \dots \bar{A}_{n-w+j-1} A_{n-w+j} \bar{A}_{n-1}] \\ & = \bar{B}_{n-w} \bar{B}_w - p^w q \sum_{j=0}^{w-2} \bar{B}_{n-2w+j} (1 - p^{w-j-1}) \\ & \leq \bar{B}_{n-w} \bar{B}_w - p^w q \sum_{j=1}^{w-1} \bar{B}_{n-w} (1 - p^{w-j}) \\ & = \bar{B}_{n-w} [\bar{B}_w - p^w q \sum_{j=1}^{w-1} (1 - p^{w-j})] \\ & = \bar{B}_{n-w} p^w q \left[ \frac{\bar{B}_w}{p^w q} + 1 - w + \sum_{i=1}^{w-1} p^i \right] \\ & = b_{n+1} \left[ \frac{1-p^w}{p^w q} + 1 - w + \sum_{i=1}^{w-1} p^i \right] \\ & = b_{n+1} \left[ \sum_{i=1}^w (p^{-i} + p^{i-1}) - w \right]. \end{aligned}$$

Taking limit and using Formula (2.2), we obtain

$$\begin{aligned} \lambda_B & = \lim_{n \rightarrow \infty} \frac{\bar{B}_{n+1}}{\bar{B}_n} = \lim_{n \rightarrow \infty} \left( 1 - \frac{b_{n+1}}{\bar{B}_n} \right) \\ & \leq 1 - \frac{1}{\sum_{i=1}^w (p^{-i} + p^{i-1}) - w}. \end{aligned}$$

(2) For any  $n \geq 2L_Q$  and  $k \geq 2$ , by the first inequality in Theorem 3.1(a) in [7],  $f_{n+k} \geq f_{n+1} \bar{Q}_{L_Q+k-2}$ . Therefore, by Formula (2.2),

$$\begin{aligned} \frac{f_{n+1}}{Q_n} & = \frac{f_{n+1}}{\sum_{i=1}^{\infty} f_{n+i}} \leq \frac{f_{n+1}}{f_{n+1} + f_{n+1} \sum_{i=0}^{\infty} \bar{Q}_{L_Q+i}} \\ & = \frac{1}{\mu_Q - L_Q + 1}, \end{aligned}$$

and so

$$\begin{aligned}\lambda_Q &= \lim_{n \rightarrow \infty} \frac{\bar{Q}_{n+1}}{Q_n} = \lim_{n \rightarrow \infty} \left(1 - \frac{f_{n+1}}{Q_n}\right) \\ &\geq 1 - \frac{1}{\mu_Q - L_Q + 1}.\end{aligned}$$

Similarly, by the second inequality in Theorem 3.1 (a) in [7],  $f_{n+1+j} \leq f_{n+1}\bar{Q}_j$  for any  $j \geq L_Q$ . Therefore,

$$\begin{aligned}\frac{f_{n+1}}{Q_n} &= \frac{f_{n+1}}{\sum_{i=1}^{\infty} f_{n+i}} \geq \frac{f_{n+1}}{\sum_{j=1}^{L_Q} f_{n+j} + f_{n+1} \sum_{i=0}^{\infty} \bar{Q}_{L_Q+i}} \\ &\geq \frac{1}{L_Q + \sum_{i=0}^{\infty} \bar{Q}_{L_Q+i}} = \frac{1}{\mu_Q},\end{aligned}$$

and so  $\lambda_Q \leq 1 - \frac{1}{\mu_Q}$ .

**THEOREM 5.2.** *Let  $Q$  be a non-uniformly spaced seed. If  $L_Q < \frac{q}{p}[(\frac{1}{p})^{w_Q-2} - 1] + 1$ , the hit probability  $Q_n$  of  $Q$  is larger than that of the consecutive seed  $B$  of the same weight when  $n$  is large enough.*

*Proof.* By Theorems 3.2 and 5.1,  $\lambda_Q \leq 1 - \frac{1}{\mu_Q} < 1 - \frac{1}{\mu_B - w_Q + 1} \leq \lambda_B$  and hence,

$$\lim_{n \rightarrow \infty} \frac{\bar{Q}_n}{B_n} = \lim_{n \rightarrow \infty} \frac{\alpha_Q \lambda_Q^n}{\alpha_B \lambda_B^n} = \lim_{n \rightarrow \infty} \frac{\alpha_Q}{\alpha_B} \left(\frac{\lambda_Q}{\lambda_B}\right)^n = 0.$$

Therefore, there exists a large integer  $N$  such that, for any  $n \geq N$ ,  $\frac{\bar{Q}_n}{B_n} < 1$  or  $Q_n > B_n$ .

**Remark.** Proposition 3.1 in [7] implies that a uniformly spaced seed  $Q$  has the same  $\lambda$  as the consecutive seed  $B$  of the same weight and for any  $n \geq L_Q$ ,  $Q_n \leq B_n$ .

## 6 Computing hit probability

**6.1 NP-hardness.** When the homologous region is uniform, that is, a Bernoulli sequence generated with probability  $p$ , there have been many exponential time algorithms to compute the hit probability of a given spaced seed [21, 19, 16, 5, 6]. However, it remains unknown whether the hit probability computation is NP-hard. The uniform distribution is structureless which makes our problem look hopelessly hard to grasp. Nevertheless, we now proceed to settle the original open question that, even under the uniform distribution, the problem of computing the hit probability of a given spaced seed is NP-hard. Our proof shares some similarity with the NP-hardness proof in [21] for the use of Lemma 6.1. However, the key idea here is a sophisticated counting in the proof after Claim 6.3 and is completely novel.

Let  $y_1$  and  $y_2$  be two real numbers between 0 and 1, represented by their binary expansions. The non-zero bits of  $y_1$  are between bit 1 to bit  $k_1$ . The non-zero bits

of  $y_2$  are between bit  $k_1 + 2$  to bit  $k_2$ . Intuitively, both  $y_1$  and  $y_2$  can be recovered easily from either  $y_1 + y_2$  or  $y_1 - y_2$ . The following simple lemma formulates this intuition and will be used several times in our proof.

**LEMMA 6.1.** *Let  $y_i = a_i \times 2^{-k_i}$  and  $0 = k_0 < k_1 < k_2 < \dots < k_n$ . If  $|a_i| < 2^{k_i - k_{i-1} - 1}$  for  $i = 1, 2, \dots, n$ , then  $y_1, y_2, \dots, y_n$  can be computed in polynomial time using  $\sum_{i=1}^n y_i$  and  $k_1, k_2, \dots, k_n$  as inputs.*

*Proof.* Let  $y'_i = 2^{-k_{i-1} - 1} + y_i$ . Then  $y'_i \geq 0$ . We only need to prove that  $y'_1, y'_2, \dots, y'_n$  can be computed in polynomial time by using  $\sum_{i=1}^n y'_i$  and  $k_1, k_2, \dots, k_n$  as inputs.

$\sum_{i=1}^n y'_i$  is represented by its binary expansion.  $k_1, \dots, k_n$  divides the bits after the decimal point into  $n$  zones, where the  $i$ -th zone consists of the bits  $k_{i-1} + 1, \dots, k_i$ . Because  $a_i < 2^{k_i - k_{i-1} - 1}$ , adding  $y'_i \geq 0$  will only change the bits in the  $i$ -th zone. Therefore, by looking at the  $i$ -th zone of the binary expansion of  $\sum_{i=1}^n y'_i$ , which is equal to  $\sum_{i=1}^n y_i + \sum_{i=1}^n 2^{-k_{i-1} - 1}$ , we can easily determine  $y'_i$ , and hence  $y_i$ .

Let  $x$  be a spaced seed which is a sequence over alphabet  $\{*, 1\}$ . Let  $|x|$  be the length of  $x$ . Let  $R$  be a random sequence over alphabet  $\{0, 1\}$  and  $|R|$  be the length of  $R$ . Recall that we use  $\mathcal{RP}(x)$  to denote its relative match position set. Let  $C_x(i)$  denote the set of required match positions of  $R$  when  $x$  hits  $R$  starting at position  $i$ . Then

$$C_x(i) = \{i + i' \mid i' \in \mathcal{RP}(x)\} := \mathcal{RP}(x) + i.$$

If we require  $x$  to hit  $R$  starting at several positions  $i_1, \dots, i_k$ , then the set of required match positions of  $R$ , denoted by  $C_x(i_1, \dots, i_k)$ , is

$$C_x(i_1, \dots, i_k) = \bigcup_{j=1}^k C_x(i_j).$$

For convenience, in this paper we sometimes say that  $x$  covers the positions in  $C_x(i_1, \dots, i_k)$  when being put at  $i_1, \dots, i_k$ . Obviously,  $|C_x(i_1, \dots, i_k)|$  may depend on the shape of  $x$  because  $C_x(i_j) \cap C_x(i_{j'})$  may be nonempty.

Let  $I = \{(i_1, i_2, \dots, i_k) \mid 1 \leq i_1 < \dots < i_k \leq |R| - |x| + 1\}$ . Let  $p$  be the probability that '1' occurs in a position in the region  $R$ . Then

$$P[x \text{ hits } R \text{ starting at } i_1, \dots, i_k] = p^{-|C_x(i_1, \dots, i_k)|}.$$

From inclusion-exclusion we know that

$$(6.7) = \sum_{k=1}^{|R|} (-1)^{k+1} \sum_{(i_1, i_2, \dots, i_k) \in I} P^{-|C_x(i_1, \dots, i_k)|}.$$

**THEOREM 6.1.** *Computing sensitivity of a spaced seed over a uniform region is NP-hard.*

*Proof.* We prove the theorem for homology level  $p = \frac{1}{2}$ . Similar proof holds for  $p = \frac{i}{j}$  for any integers  $i < j$ . We will reduce 3-Set-Cover to the hit probability computation. Suppose we have a set  $X = \{1, 2, \dots, n\}$ , and  $m$  size-3 sets  $X_i \subset X$ ,  $i = 1, 2, \dots, m$ . The 3-Set-Cover problem asks whether there are  $K$  of the  $m$  subsets,  $X_{i_1}, \dots, X_{i_K}$ , such that  $X = \bigcup_{k=1}^K X_{i_k}$ . It is well-known to be NP-hard [11].

The core of our reduction is a length- $n$  string  $s_i$  for each set  $X_i$  ( $i = 1, \dots, m$ ), in which  $s_i[j] = 1$  if and only if  $j \in X_i$ . Then we put these  $s_k$  into a single string

$$S_1 = s_1 1^{n(m^2+1)} s_2 1^{n(m^2+2)} s_3 \dots 1^{n(m^2+m-1)} s_m.$$

For any  $r_1, \dots, r_k$ , there are  $k$  positions  $i_1, \dots, i_k$ , such that putting  $k$  copies of  $S_1$  at positions  $i_1, \dots, i_k$  will align the corresponding  $s_{r_1}, \dots, s_{r_k}$  together. If  $X_{r_1}, \dots, X_{r_k}$  cover  $X$ , then the overlap of  $s_{r_1}, \dots, s_{r_k}$  will cover  $n$  positions. Otherwise, they cover fewer than  $n$  positions. Our proof will exploit the difference made by this design: when there are  $K$  sets that cover  $X$ , there is at least one more combination of  $(i_1, \dots, i_k)$  such that the  $k$  copies of  $S_1$  cover some more bits.

It is also obvious that  $|S_1| = nm(m^2 - \frac{m}{2} + \frac{1}{2})$ . Let  $N = |S_1|$ . Let  $K_2, K_3, K_4, K_5$  be sufficiently large numbers to be determined later. The rest of the reduction consists of the following components:

1.  $S_2 = (1^N S_1)^{K_2} 1^N$ . By repeating  $S_1$  many times, we “amplify” the above mentioned difference. Also, the  $1^N$  between copies of  $S_1$  simplify the analysis at the boundary of  $S_1$ .
2.  $S_3 = (1^N 0^N)^{K_3}$ . When  $k$  copies of  $S_3$  are put at  $k$  different positions  $1 \leq i_1 < \dots < i_k \leq N$ , the coverage is  $(N + i_k - i_1)K_3$ . That is, the coverage only depends on the two farthest positions. Again,  $K_3$  is to amplify the situation.
3.  $S_4 = (10^{n-1})^{K_4}$ . When two copies of  $S_4$  are put at two different positions  $i_1 < i_2$ , if  $i_2 = i_1 \bmod n$ , the coverage is  $K_4 + (i_2 - i_1)/n$ . However, if  $i_2 \neq i_1 \bmod n$ , then the coverage is  $2K_4$ , which is significantly higher than  $K_4 + (i_2 - i_1)/n$ .  $S_4$  will allow us to focus on the cases where  $s_i$  overlaps  $s_j$  either completely or completely not.
4.  $S_5 = (10^N)^{K_5}$ . Putting  $S_5$  at  $k$  different positions  $1 \leq i_1 < \dots < i_k \leq N$  will cover  $kK_5$  positions. This will allow us to examine different values of  $k$  separately.

5. Finally, we let our seed be  $x = S_2 0^N S_3 0^N S_4 0^N S_5$ . The random region  $R$  has length  $|x| + N - 1$  and identity level  $\frac{1}{2}$ .

We will show that the accurate computation of the hit probability of  $x$  at  $R$  will give a polynomial time algorithm to the original 3-Set-Cover problem.

**CLAIM 6.1.** *Let  $K_5 = 2 \times |S_2 0^N S_3 0^N S_4 0^N|$ . Then*

$$P_k = \sum_{1 \leq i_1 < \dots < i_k \leq N} 2^{-|C_x(i_1, \dots, i_k)|}$$

*can be recovered from  $P(x \text{ hits } R)$ .*

*Proof.* From (6.7),  $P[x \text{ hits } R] = \sum_{k=1}^N (-1)^{k+1} P_k$ . Because of the existence of  $S_5$ ,  $C_x(i_1, \dots, i_k) > kK_5$ . On the other hand,

$$C_x(i_1, \dots, i_{k-1}) < (k-1)K_5 + \frac{K_5}{2} = (k - \frac{1}{2})K_5.$$

Moreover, there are  $\binom{N}{k} < 2^N$  possible  $(i_1, \dots, i_k)$  and  $N < \frac{K_5}{2} - 1$ . From Lemma 6.1, the claim is true.

**CLAIM 6.2.** *Let  $K_4 = 2 \times |S_2 0^N S_3 0^N|$ . Let*

$$\begin{aligned} \mathcal{I}' &= \{(i_1, \dots, i_k) \mid 1 \leq i_1 < \dots < i_k \leq N\} \\ \mathcal{I} &= \{(i_1, \dots, i_k) \in \mathcal{I}' \mid i_1 = \dots = i_k \bmod n\}. \end{aligned}$$

*Then*

$$x_k = \sum_{(i_1, \dots, i_k) \in \mathcal{I}} 2^{-|C_x(i_1, \dots, i_k)|}$$

*can be computed accurately from  $P_k$ .*

*Proof.* For each  $i_1, \dots, i_k$  that  $i_1 = \dots = i_k \bmod n$  is not true, the coverage caused by  $S_4$  only is at least  $2K_4$ . The coverage caused by  $S_5$  is equal to  $kK_5$ . As a result,

$$C_x(i_1, \dots, i_k) > kK_5 + 2K_4.$$

On the other hand, for each  $i_1, \dots, i_k$  that  $i_1 = \dots = i_k \bmod n$  is true, the coverage caused by  $S_4$  is at most  $K_4 + \frac{N}{n}$ ; and the coverage caused by  $S_2 0^N S_3 0^N$  is at most  $\frac{K_4}{2}$ . As a result,

$$C_x(i_1, \dots, i_k) < kK_5 + K_4 + \frac{N}{n} + \frac{K_4}{2} < kK_5 + \frac{5}{3}K_4.$$

Moreover, there are  $\binom{N}{k} < 2^N$  possible  $(i_1, \dots, i_k)$  and  $N < \frac{K_4}{3} - 1$ . From Lemma 6.1, the claim is true.

CLAIM 6.3. Let  $K_3 = 2 \times |S_2 0^N|$ . Let

$$\mathcal{I}_l = \{(i_1, \dots, i_k) \in \mathcal{I} \mid i_k - i_1 = l\}.$$

Then

$$x_{k,l} = \sum_{(i_1, \dots, i_k) \in \mathcal{I}_l} 2^{-|C_x(i_1, \dots, i_k)|}$$

can be computed accurately from  $x_k$ .

*Proof.* From the definition of  $x_k$  and  $x_{k,l}$ , we know that  $x_k = \sum_{l=1}^N x_{k,l}$ . For any  $l$ , the coverage caused by  $S_3$  is  $(N+l)K_3$ . Together with the coverage caused by  $S_4$  and  $S_5$ , for  $(i_1, \dots, i_k) \in \mathcal{I}_l$ ,

$$C_x(i_1, \dots, i_k) > kK_5 + K_4 + (N+l)K_3.$$

On the other hand, for  $(i_1, \dots, i_k) \in \mathcal{I}_{l-1}$ , the coverage caused by  $S_2 0^N$  is at most  $\frac{K_3}{2}$ . Then

$$\begin{aligned} & C_x(i_1, \dots, i_k) \\ < & kK_5 + (K_4 + \frac{N}{n}) + (N+l-1)K_3 + \frac{K_3}{2} \\ < & kK_5 + K_4 + (N+l)K_3 - \frac{K_3}{3}. \end{aligned}$$

Moreover, there are  $\binom{N}{k} < 2^N$  possible  $(i_1, \dots, i_k)$  and  $N < \frac{K_3}{3} - 1$ . From Lemma 6.1, the claim is true.

Next let us examine the relationship between  $x_{k,l}$  and the answer of the 3-set-cover problem. Given  $k, l$ , for any  $(i_1, \dots, i_k) \in \mathcal{I}_l$ , because of the definition of  $\mathcal{I}_l$ , there are three cases if  $x$  is put at positions  $i_1, \dots, i_k$ .

**Case 1:** There are no  $n$  consecutive columns where  $s_{r_1}, \dots, s_{r_k}$  aligned together.

In this case, the coverage caused by  $S_2$  is  $(2K_2 + 1)N + l$ , and therefore

$$\begin{aligned} & C_x(i_1, \dots, i_k) \\ = & kK_5 + K_4 + \frac{l}{n} + (N+l)K_3 + (2K_2 + 1)N + l \end{aligned}$$

**Case 2:** There are  $n$  consecutive columns where  $s_{r_1}, \dots, s_{r_k}$  are aligned together. Furthermore,  $s_{r_1}, \dots, s_{r_k}$  cover all the  $n$  positions. (Correspondingly,  $X_{r_1}, \dots, X_{r_k}$  cover  $X$ ). It is easy to see that  $C_x(i_1, \dots, i_k)$  is the same as in Case 1.

**Case 3:** There are  $n$  consecutive columns where  $s_{r_1}, \dots, s_{r_k}$  aligned together, respectively. But  $s_{r_1}, \dots, s_{r_k}$  do not cover all the  $n$  positions. (Correspondingly,  $X_{r_1}, \dots, X_{r_k}$  do not cover  $X$ ). Then the coverage caused by  $S_2$  is no more than  $(2K_2 + 1)N + l - K_2$ . Hence,

$$\begin{aligned} C_x(i_1, \dots, i_k) \leq & kK_5 + K_4 + \frac{l}{n} + (N+l)K_3 \\ & + (2K_2 + 1)N + l - K_2. \end{aligned}$$

Let  $K_2 = 2N$ . Because the  $K_2$  difference between case 3 and the other two cases, by checking  $x_{k,l}$  at bits between  $K_5k + K_4 + \frac{l}{n} + (N+l)K_3 + (2K_2 + 1)N + l - K_2 + 1$  and  $K_5k + K_4 + \frac{l}{n} + (N+l)K_3 + (2K_2 + 1)N + l$ , we can get the total number of  $(i_1, \dots, i_k) \in \mathcal{I}_l$  for the *first two* cases.

It turns out that the total number of  $(i_1, \dots, i_k) \in \mathcal{I}_l$  for the *first* case can be computed easily in a different way, to be described in the following. Therefore, by comparing the numbers for the first two cases and for the first case, we will be able to know whether case 2 exists. The 3-set-cover can then be answered.

For a fixed  $i_1$ , the number of different  $(i_1, \dots, i_k)$  such that  $i_1 = \dots = i_k \pmod n$  and  $i_k - i_1 = l$  is  $\binom{\frac{l}{n}-1}{k-2}$ . And  $i_1$  can take values from 1 to  $N - l$ . Therefore,  $|\mathcal{I}_l| = \binom{\frac{l}{n}-1}{k-2} \times (N - l)$ .

Recall that  $l$  stands for  $i_k - i_1$  for  $(i_1, \dots, i_k) \in \mathcal{I}_l$ . For some values of  $l$ , there exists a pair of  $r_1$  and  $r_k$  such that putting  $x$  at  $i_1$  and  $i_k$  will align  $s_{r_1}$  and  $s_{r_2}$  together. Because of the design  $S_1$ ,  $r_1$  and  $r_k$  can be uniquely determined by  $l$  and can be easily computed in polynomial time. Then for a fixed  $i_1$ , the number of  $(i_1, \dots, i_k) \in \mathcal{I}_l$  that can align  $s_{r_1}, s_{r_2}, \dots, s_{r_k}$  together is equal to the number of  $(r_1, \dots, r_k)$  such that  $r_1 < \dots < r_k$ , which is  $\binom{r_1 - r_k - 1}{k-2}$ . Also because  $i_l$  has  $N - l$  choices, the number of  $(i_1, \dots, i_k)$  in Case 1 is

$$(6.8) \quad |\mathcal{I}_l| = \binom{r_1 - r_k - 1}{k-2} \times (N - l).$$

For some other values of  $l$ , there is no pair of  $r_1$  and  $r_k$  such that  $s_{r_1}$  of  $x$  at  $i_1$  and  $s_{r_k}$  of  $x$  at  $i_k$  are aligned together. Then, the number of combinations in Case 1 is equal to

$$(6.9) \quad |\mathcal{I}_l| = \binom{\frac{l}{n} - 1}{k-2} \times (N - l).$$

That is, for given  $k$  and  $l$ , by using either (6.8) or (6.9), we can determine the number of  $(i_1, \dots, i_k)$  in Case 1. On the other hand, from the previous discussion, if  $P[x \text{ hits } R]$  is known, we were able to determine the total number of  $(i_1, \dots, i_k)$  in Case 1 and 2. Therefore, we are able to say whether there is an  $(i_1, \dots, i_k)$  in Case 2. By examining all  $l$ , the set cover question is answered.

The key idea in the above proof is that the probability of hits starting at  $(i_1, \dots, i_k)$  can only affect a limited range of the binary expansion of the hit probability  $P[x \text{ hits } R]$ . This property still holds if we change the identity level from  $\frac{1}{2}$  to  $\frac{i}{j}$  for any integers  $i < j$  and change the binary expansion of the hit probability to a base  $j$  expansion. As a result, by slightly changing the proof, the NP-hardness follows for identity level  $\frac{i}{j}$  for all  $i < j$ .

*Remark.* What about the complexity of *finding* the optimal seed? We have mentioned earlier that it is NP-hard to find the optimal seed in some distribution [19]. Can we similarly improve this result under uniform distribution? The answer is “not likely”, as now the problem is defined by the language:  $L_p = \{(1^L, 1^w, Q) \mid w \leq L\}$ , where  $Q$  is the optimal seeds of length  $L$  and weight  $w$  on a uniform region of homology level  $p$ . Such a set is sparse and a sparse set being NP-hard implies NP = P [22].

## 6.2 Computing the asymptotic hit probability.

Now we know that computing the hit probability of a spaced seed is NP-hard. This justifies that the exponential time algorithms used in all the papers [21, 16, 6, 7, 5, 19, 33]. However, the time complexities of all these algorithms also depend on the homologous region length, and therefore not useful for comparing the hit probabilities of two seeds asymptotically as discussed in Section 5.

In this section, we extend the argument of Buhler *et al.* [6], but using a different line of argument, to give an exponential time algorithm to compute the hit probability of a spaced seed, independent of homologous region length. Thus this also gives an effective method to compare the asymptotic sensitivity of two seeds.

A positive matrix is a matrix whose entries are strictly greater than 0. Then, we have

LEMMA 6.2. ([23, 4]) *Let  $A = (a_{i,j})$  be a positive matrix. Then,*

1. *A has a unique largest eigenvalue  $\lambda_1$  in  $[\min_i \sum_j a_{ij}, \max_i \sum_j a_{ij}]$ . Furthermore, the eigenvector associated with  $\lambda_1$  can be normalized so that all its components are positive.*
2. *Let  $M = \max_{i,j} a_{ij}$ ,  $m = \min_{i,j} a_{ij}$ , and  $K = \max_{i,j,k,l} \sqrt{\frac{a_{ij}a_{kl}}{a_{il}a_{kj}}}$ . Then, other eigenvalues  $\lambda$  of A satisfy the inequality below:*

$$(6.10) \quad |\lambda| \leq \frac{K-1}{K+1} \lambda_1 \leq \frac{M-m}{M+m} \lambda_1.$$

THEOREM 6.2. *Given a spaced seed  $Q$  and homology level  $p$ . The asymptotic hit probability on a homologous region  $R$  can be computed in time exponentially proportional to the length  $L_Q$  of  $Q$ , independent of the length of  $R$ .*

The proof will be provided in the full versions of the paper. The following corollary answers an open question raised in [7].

COROLLARY 6.1. *Given two seeds  $Q_1$  and  $Q_2$  of the same length  $L$ , their limiting sensitivities are well-defined by their largest eigenvalues of their transition matrices [6], and they can be effectively compared in exponential time in  $L$ .*

## 6.3 Efficient approximation of the hit probability of a seed.

Authors of [19] proposed a simple random sampling algorithm for computing the hit probability. The algorithm guarantees the absolute error to be small. However, this does not guarantee an approximation ratio as the hit probability can be very small for lower identity level and higher seed weight. In this case, a small absolute error may cause a very bad approximation ratio. To guarantee an approximation ratio, the time complexity of such algorithm has to depend on homology level and seed weight and can be very high. In this section, we give an efficient and practical polynomial time approximation algorithm, guaranteeing arbitrarily good performance ratio with high probability, independent of homology level and seed weight.

Let  $p$  be the identity level of  $R$ . That is,  $R[i] = 1$  with probability  $p$  for  $0 \leq i < L$ , where  $L$  is the length of  $R$ . Let  $N$  be a large number. Our algorithm is the following.

**Algorithm WiseSample**

$n_j \leftarrow 0$  for  $j = 1, \dots, L - L_Q$ ;  
Repeat  $N$  times  
  Let  $R[i] = 1$  for  $i \in \mathcal{RP}(Q)$ ;  
  For  $i \notin \mathcal{RP}(Q)$ , let  $R[i] = 1$  with probability  $p$ .  
  For  $i = 1, \dots, L - L_Q$   
    if  $Q$  does not hit  $R[1, i + L_Q - 1]$   
       $n_i \leftarrow n_i + 1$ .  
Output  $p^{w_Q} (1 + N^{-1} \sum_{j=1}^{L-L_Q} n_j)$ .

The proof of the following theorem will be provided in the full versions of the paper.

THEOREM 6.3. *Let the hit probability of  $Q$  be  $x$ . For any  $\epsilon > 0$ , let  $N = \left\lceil \frac{6L^2 \log L}{\epsilon^2} \right\rceil$ . Then with high probability, Algorithm WiseSample outputs value  $y$  such that  $|y - x| \leq \epsilon x$ .*

## 7 Acknowledgment

L. Zhang would like to thank K.P. Choi, F. Preparata and Y. Kong for valuable discussions on different aspects of spaced seeds. In particular, Theorem 3.1 is obtained by discussion with Y. Kong. Our research is supported by the NSERC of Canada, CITO’s Champion of Innovation Program, the PREA award, the Killam Research Fellowship, the Canada Research Chair Program, Singapore BMRC research grant, and NUS AFR grant.

## References

- [1] S.F. Altschul *et al.*, Basic local alignment search tool, *J. Mol. Biol.* **215**(1990), pp. 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), pp. 3389-3402.
- [3] N. Balakrishnan and M.V. Koutras, *Runs and Scans with Applications*, John Wiley & Sons, U.S.A., 2002.
- [4] G. Birkhoff, Extension of Jentzsch's theorem, *Trans. Amer. Math. Soc.* **85**(1957), pp. 219-227.
- [5] B. Brejova, D. Brown, and T. Vinař, Optimal spaced seeds for homologous coding regions. *J. Bioinf. and Comp. Biol.* **1**(2004), pp. 595-610. Early version appeared in CPM 2003.
- [6] J. Buhler, U. Keich, and Y. Sun, Designing seeds for similarity search in genomic DNA. *Proc. 7th Annual Int'l Conf. on Comput. Mol. Biol.* (RECOMB'03), pp. 67-75, Berlin, Germany.
- [7] K.P. Choi, and L. Zhang, Sensitivity analysis and efficient method for identifying optimal spaced seeds, *J. Comput. Sys. Sci.*, **68**(2004), pp. 22-40.
- [8] K.P. Choi, F. Zeng, and L. Zhang, Good Spaced Seeds for Homology Search, *Bioinformatics* **20**(2004), pp. 1053-1059.
- [9] W. Feller, *An introduction to Probability Theory and its Applications*, Vol. I (3rd edition), Wiley, New York.
- [10] F. Nicolas, and E. Rivals, Hardness of Optimal Spaced Seed Designi, *Proc. 16th Annual Symposium Combinatorial Pattern Matching* (CPM'05), 2005, pp. 144-155.
- [11] M. Garey, and D. Johnson, *A guide to the theory of NP-completeness*, W.H. Freeman, 1979.
- [12] V. Gotea, V. Veeramachaneni, and W. Makalowski, Mastering seeds for genomic size nucleotide BLAST searches, *Nucleic Acids Res.* **31**(2003), pp. 6935-6941.
- [13] L.J. Guibas and A.M. Odlyzko, String overlaps, pattern matching, and nontransitive games, *J. of Combin. Theory* (series A) **30**(1981), pp. 183-208.
- [14] Int'l Mouse Genome Sequencing Consortium, Initial sequencing and comparative analysis of the mouse genome. *Nature* **409**(2002), pp. 520-562.
- [15] P. Jacquet and W. Szpankowski, Analytic Approach to Pattern Matching, In *Applied Combinatorics on Words* (Editor: M. Lothaire), Cambridge Press, 2005.
- [16] U. Keich, M. Li, B. Ma, and J. Tromp, On Spaced Seeds for Similarity Search, *Discrete Appl. Math.* **3**(2004), pp. 253-263.
- [17] G. Kucherov, L. Noe, and Y. Ponty, Estimating seed sensitivity on homogeneous alignments, In *Proc. IEEE 4th Symp. on Bioinformatics and Bioengineering*, Taiwan, 2004, pp. 387-394.
- [18] M. Li, B. Ma, and L. Wang, On the Closest String and Substring Problems, *Journal of the ACM* **49**(2002), pp. 157-171.
- [19] M. Li, B. Ma, D. Kisman, and J. Tromp, PatternHunterII: highly sensitive and fast homology search. *J. Bioinformatics and Comput. Biol.* **2**(2004), pp. 417-440.
- [20] D.J. Lipman and W.R. Pearson, Rapid and sensitive protein similarity searches, *Science* **227**(1985), pp. 1435-1441.
- [21] B. Ma, J. Tromp, and M. Li, PatternHunter: faster and more sensitive homology search, *Bioinformatics*, **18**(2002), pp. 440-445.
- [22] S.R. Mahaney, Sparse complete sets for NP: solution of a conjecture of Berman and Hartmanis, *J. Comput. System Sci.* **25**(1982), pp. 130-143.
- [23] H. Minc, *Nonnegative matrices*, John Wiley and Sons, New York, 1988.
- [24] National Center for Biotechnology Information. Growth of GenBank, 2002.
- [25] S.B. Needleman and C.D. Wunsch, A general method applicable to the search for similarities in the amino acid sequence of two proteins, *J. Mol. Biol.* **48**(1970), pp. 443-453.
- [26] P. Nicodeme, B. Salvy, and P. Flajolet, Motif Statistics. *Lecture Notes in Computer Sciences*, vol. 1643, 1999, pp. 194-211.
- [27] F.P. Preparata, L. Zhang, and K.P. Choi, Quick, practical selection of effective seeds for homology search, To appear in *J. Comput. Biology*, 2005.
- [28] G. Reinert, S. Schbath, and M. Waterman, Probabilistic and statistical properties of words: An overview, *J. Comput. Biol.* **7**(2000), 1-46.
- [29] S. Schwartz *et al.* Human-Mouse alignment with BLASTZ, *Genome Res.***13**(2003), pp. 103-107.
- [30] T.F. Smith and M.S. Waterman, Identification of common molecular subsequences, *J. Mol. Biol.* **147** (1981), pp. 195-197.
- [31] A.D. Solov'ev, A combinatorial identity and its application to the problem concerning the first occurrences of a rare event, *Theory of Probab. and Appl.* **11**(1966), pp. 276-282.
- [32] Y. Sun and J. Buhler, Designing multiple simultaneous seeds for DNA similarity search, in *Proc. of RECOMB'04*, 2004, pp. 76-85.
- [33] J. Xu, D. Brown, M. Li, and B. Ma, Optimizing multiple spaced seeds for homology search, In *Proc. of CPM'04*, LNCS, vol. 3109, pp. 47-58. Final version to appear in *J. Comp. Biol.*
- [34] I.-H. Yang *et al.* Efficient Methods for Generating Optimal Single and Multiple Spaced Seeds, In *Proc. IEEE 4th Symp. on Bioinformatics and Bioengineering*, Taiwan, 2004, pp. 411-418.
- [35] Z. Zhang, S. Schwartz, L. Wagner, and W. Miller, A greedy algorithm for aligning DNA sequences. *J. Comput. Biology.* **7**(2000), pp. 203-214.