Available at
www.ElsevierMathematics.com
POWERED BY SCIENCE DIRECT®

Discrete Applied Mathematics 138 (2004) 253–263

DISCRETE APPLIED MATHEMATICS

www.elsevier.com/locate/dam

On spaced seeds for similarity search

Uri Keich[a,*], Ming Li[b], Bin Ma[c], John Tromp[d]

[a] *Computer Science & Engineering Department, University of California, San Diego, CA 92093, USA*
[b] *Bioinformatics Lab, Computer Science Department, University of California, Santa Barbara, CA 93106, USA*
[c] *Computer Science Department, University of Western Ontario, London Canada N6A 5B8*
[d] *CWI, P.O. Box 94079 1090 GB Amsterdam, Netherlands*

Received 16 June 2002; received in revised form 8 May 2003; accepted 30 June 2003

Abstract

Genomics studies routinely depend on similarity searches based on the strategy of finding short seed matches (contiguous k bases) which are then extended. The particular choice of the seed length, k, is determined by the tradeoff between search speed (larger k reduces chance hits) and sensitivity (smaller k finds weaker similarities). A novel idea of using a single deterministic optimized spaced seed was introduced in Ma et al. (Bioinformatics (2002) 18) to the above similarity search process and it was empirically demonstrated that the optimal spaced seed quadruples the search speed, without sacrificing sensitivity. Multiple, randomly spaced patterns, spaced q-grams, and spaced probes were also studied in Califano and Rigoutsos (Technical Report, IBM T.J. Watson Research Center (1995), Burkhardt, Kärkkäinen, CPM (2001), and Buhler, Bioinformatics 17 (2001) 419) and in other applications [(RECOMB (1999) 295, RECOMB (2000) 245)]. They were all found to be better than their contiguous counterparts. In this paper we study some of the theoretical and practical aspects of optimal seeds. In particular we demonstrate that the commonly used contiguous seed is in some sense the worst one, and we offer an algorithmic solution to the problem of finding the optimal seed.

MSC: 92D20

Keywords: Gapped seeds; Seeded alignment; BLAST; Similarity search

[*] Computer Science, Cornell University, 4130 Upson Hall, Ithaca, NY 14853, USA. Tel.: 607-2550983; fax: 607-2554428.
E-mail address: keich@cs.cornell.edu (U. Keich).

0166-218X/$ - see front matter © 2003 Elsevier B.V. All rights reserved.
doi:10.1016/S0166-218X(03)00382-2

1. Introduction

Today, in the post-genomics era [8,16], the most common computer task performed in molecular biology labs is similarity search. That is, to compare one DNA sequence against another, or to a database, to find any similar regions between them. Many programs have been developed for the task. These include FASTA [9], SIM [7], the Blast family [1,2,6,15,18] SENSEI [14], and recently Pattern Hunter [10]. The popularity of such practice can be seen from the fact that the original Blast paper [1] is the most referred scientific paper of the last decade: cited over 10,000 times. Almost all these programs, including NCBI's widely used Blastn, use the simple strategy of first finding short exact "seed" matches (hits), which are then extended into longer alignments. This approach to similarity search exhibits a key tradeoff: increasing seed size decreases sensitivity, whereas decreasing seed size slows down the computation.

In order to alleviate this problem, a novel idea of using a single deterministic and optimized spaced seed was introduced in [10]. Using this spaced seed yields, a similarity search which is as sensitive as the one based on the naive, contiguous seed but produces 4 times fewer chance hits and hence is about 4 times faster [10]. Multiple, possibly randomized, spaced patterns, spaced q-grams, and spaced probes were also studied by Califano and Rigoutsos [5], Burkhardt and Kärkkäinen [4], Buhler [3], and Preparata et al. [11], Preparata and Upfal [12] in other applications. They were all found to be significantly better than their contiguous counterparts.

At first glance it seems surprising that spaced seeds could have an advantage over contiguous ones. Indeed, in a region of similarity $0 \leqslant p \leqslant 1$, the probability of a match between a pair of W positions is p^W for any seed. Since a contiguous seed can fit in a given region in more ways than a, necessarily longer, spaced seed, the expected number of matches of the contiguous seed is higher than that of any spaced seed in the same region [10]. In this paper we study some of the theoretical and practical aspects of optimal seeds as defined in [10]. In particular we demonstrate that a contiguous seed is in some sense the worst one, and we offer an algorithmic solution to the problem of finding the optimal seed.

2. Mathematical formulation

The traditional seed that is used in programs such as Blast consists of k consecutive positions. That is, the program looks for a word of length k which appears in each of the studied pair of sequences. Ma et al. [10] empirically observed that better results can be obtained if we are allowed to space the preserved k positions. The authors of [10] call the specific pattern of the matching positions a "model seed" (or just "seed") and describe it by a 0-1 string, where the 1s correspond to the matching positions. For example, if we use the seed 1110111, then the pair of words `actgact` and `acttact` is a seed match, and so is the pair `actgact` and `actgact`. The number of 1s is called the "weight" of the seed and it has direct impact on the sensitivity as well as on the cost of the similarity search. The length, or "span" of the seed is its overall length, or the number of 0s plus the number of 1s in the seed.

The following notations are adopted from Burkhardt and Kärkkäinen [4]. By the *shape*, Q, of the seed we mean the relative positions of the 1s in the seed. In other words, the shape is a set of non-negative integers including 0. With this definition, the weight of Q [1] is simply $|Q|$, and its span is $s(Q) = \max Q + 1$. For any integer i and shape Q, the positioned shape $i+Q$ is the set $\{i+j : j \in Q\}$. Let, $i+Q=\{i_0, i_1, \ldots, i_{W-1}\}$, where $i = i_0 < i_1 < \cdots < i_{W-1}$, and let $S = s_0 s_1 \ldots s_{L-1}$ be a string. For $0 \leqslant i \leqslant L - s(Q)$, the *$Q$-gram* at position i in S, denoted by $S[i+Q]$, is the string $s_{i_0} s_{i_1} \ldots s_{i_{W-1}}$. Two strings S and S' have a common Q-gram at position i if $S[i+Q] = S'[i+Q]$.

Example 1. Let $Q = \{0, 1, 3, 6\}$ be a shape. Then, Q is the seed 1101001. Its weight is $|Q| = 4$ and its span $s(Q) = 7$. The string $S = \mathtt{ACGGATTAC}$ has three Q-grams: $S[0+Q] = s_0 s_1 s_3 s_6 = \mathtt{ACGT}$, $S[1+Q] = \mathtt{CGAA}$ and $S[2+Q] = \mathtt{GGTC}$.

The problem presented in [10] is finding the optimal shape (seed) of a given weight, W, for detecting identities in a region of similarity level p. More precisely, assuming we have two (aligned) strings S and S' of length L such that the events $E_i = \{S[i] = S'[i]\}$ are mutually independent and $\mathrm{P}(E_i) = p$, what is the shape Q which maximizes the *sensitivity*

$$\mathrm{P}(\exists i \in \{0, 1, \ldots, L - s(Q)\} \text{ with } S[i+Q] = S'[i+Q]).$$

By translating a match at position i, $S[i] = S'[i]$, to the digit 1, and a mismatch to 0, this problem is transformed to the following equivalent one: Let S be a random sequence of iid Bernoulli random variables with $\mathrm{P}(S[i] = 1) = p$. Let $\mathbf{1}_W = 11\ldots1$ denote the string of W consecutive 1s. If for some $0 \leqslant i \leqslant L - s(Q), S[i+Q] = \mathbf{1}_W$, then we say that Q *hits* S (at i). We look for the shape Q, of weight W which maximizes $\mathrm{P}(Q \text{ hits } S)$. In this paper we suggest a practical approach for solving this problem. We also identify conditions under which we can prove that the naive, contiguous, seed $\tilde{Q} = \{0, 1, \ldots, W-1\}$ is the worst possible seed: any (equally weighted) spaced seed will do better.

3. Toward proving the advantage of spaced seeds

Even in the framework of our model spaced seeds are not always better than the naive seed which consists of a contiguous block of 1s. In general, the sensitivity of the seed varies not only with the seed itself but it is also a function of the similarity level, p, and of the length of the similarity region, L. Consider for example looking for a weight W seed in a region of length $L = W + 1$. Clearly, the shorter, contiguous seed is the most sensitive for this problem. Similarly, the authors of [10] report that the optimal seed of weight 11 for a 64 random bits string, [2] S, is 111010011001010111. While this is certainly the case for any practical value of p, it should be noted that there are some ps for which the naive, or contiguous, seed will do better than the

[1] In what follows we identify the seed with its shape.

[2] More precisely, S is made of a sequence of 64 independent Bernoulli trials with $\mathrm{P}(1) = p$.

otherwise optimal spaced seed. One reason for this is that for a given string length the two seeds or shapes are not on equal footing as the spaced seed has a longer span so it can match only 64–17 possible substrings, or words, of S while the naive motif gets seven "extra attempts".[3]

Given these examples, it is clear that in order to make any mathematical statement regarding the advantage of spaced seeds we have to restrict the setup. For example, it turns out that if we "level the playing field" between a spaced seed and the naive one, then the former is a better seed independently of p and L. More precisely, let Q be the shape of a spaced seed. Then for any increasing sequence of indices $0 \leqslant i_0 < i_1 < \cdots < i_{n-1}$, let A_j be the event $A_j = \{S[i_j + Q] = \mathbf{1}_W\}$, or in other words, the seed matches the i_jth word of S. Similarly, for $0 \leqslant j < n$, let $\tilde{A}_j = \{S[j + \tilde{Q}] = \mathbf{1}_W\}$, or the jth word of S matches the naive (all-1) seed. Note that A_j is defined as a match (of Q) at i_j, whereas $\tilde{A}_j$ is defined as match (of $\tilde{Q}$) at j.

Claim 1. *For any increasing sequence of indices* $0 \leqslant i_0 < i_1 < \cdots < i_{n-1} \leqslant |S| - s(Q)$,

$$\mathrm{P}\left(\bigcup_{j<n} A_j\right) \geqslant \mathrm{P}\left(\bigcup_{j<n} \tilde{A}_j\right). \tag{1}$$

Moreover, for $i_j = j, n \geqslant 2$, (1) *holds with strict inequality.*

Corollary 1. *For any n, a spaced seed is more likely to match one of the first n words of S than the contiguous seed is.*

Corollary 2. *Assume S is an infinite string and let τ_Q be the first position i for which $S[i + Q] = \mathbf{1}_W$, or the first time the seed hits. Then,* $E[\tau_Q] < E[\tau_{\tilde{Q}}]$.[4]

Proof of corollaries. The first corollary is simply the claim specialized to $i_j = j$. As for the second corollary, letting $i_j = j$,

$$\begin{aligned} E[\tau_Q] &= \sum_{k=0}^{\infty} \mathrm{P}(\tau_Q > k) \\ &= \sum_{k=0}^{\infty} (1 - \mathrm{P}(\overset{k}{\underset{j=0}{\cup}} A_j)) \\ &< \sum_{k=0}^{\infty} (1 - \mathrm{P}(\overset{k}{\underset{j=0}{\cup}} \tilde{A}_j)) \\ &= E[\tau_{\tilde{Q}}]. \qquad \square \end{aligned}$$

[3] One might hope that for a circular string S a spaced seed is always better than the contiguous one; unfortunately, even for a circular string that is not necessarily the case. For example, one can show that the probability that the seed 110011 is detected in a circular region of length 16 with $p = 0.96$ is 0.9999749 which is less than 0.9999756 for the contiguous seed 1111.

[4] Note that $E\tau_{\tilde{Q}} = \sum_{j=1}^{W} p^{-j} - W$ (e.g. [13]).

Proof of claim. We first prove the weak inequality (1). Considering the indices $i_j - i_0$ we can assume without loss of generality that $i_0 = 0$ and we prove the claim by induction on n. For $n = 1$,

$$\mathrm{P}(A_0) = p^W = \mathrm{P}(\tilde{A}_0).$$

Assuming now that the claim holds for all $n \leqslant N$ we show it holds for $n = N+1$. For $0 \leqslant k < W$ let $\tilde{E}_k = \{S[0] = S[1] = \cdots = S[k-1] = 1, S[k] = 0\}$ and $\tilde{E}_W = \{S[0+\tilde{Q}] = \mathbf{1}_W\}$. Similarly, with $Q[j]$ denoting the jth offset index of Q, let $E_k = \{S[Q[0]] = S[Q[1]] = \cdots = S[Q[k-1]] = 1, S[Q[k]] = 0\}$ and $E_W = \{S[Q] = \mathbf{1}_W\}$. Clearly, both $\{\tilde{E}_k\}$ and $\{E_k\}$ $(0 \leqslant k \leqslant W)$ are partitions of the sample space and since for all k, $\mathrm{P}(E_k) = \mathrm{P}(\tilde{E}_k)$, it suffices to show that for any $0 \leqslant k \leqslant W$,

$$\mathrm{P}\left(\bigcup_{j=0}^{N} A_j | E_k\right) \geqslant \mathrm{P}\left(\bigcup_{j=0}^{N} \tilde{A}_j | \tilde{E}_k\right). \tag{2}$$

Clearly, for $k = W$ both sides of (2) equal 1. For $k < W$ note that $(\cup_{j \leqslant k} \tilde{A}_j) \cap \tilde{E}_k = \emptyset$ and that $\{\tilde{A}_{k+1}, \tilde{A}_{k+2}, \ldots, \tilde{A}_N\}$ are mutually independent of E_k, thus

$$\mathrm{P}\left(\bigcup_{j=0}^{N} \tilde{A}_j | \tilde{E}_k\right) = \mathrm{P}\left(\bigcup_{j=k+1}^{N} \tilde{A}_j\right). \tag{3}$$

The analysis of the first term in (2) is slightly more involved. Fix a $k \in \{0, \ldots, W-1\}$ and note that at most $k+1$ of the events A_j satisfy $A_j \cap E_k = \emptyset$. Indeed, $A_j \cap E_k = \emptyset$ if and only if $Q[k] \in i_j + Q$ if and only if $Q[k] - i_j \in Q$, which leaves at most $k+1$ choices for i_j. Thus, there exists indices $0 < m_{k+1} < m_{k+2} < \cdots m_N \leqslant N$ such that $A_{m_j} \cap E_k \neq \emptyset$. Since the occurrence of E_k implies that $S[Q[0]] = S[Q[1]] = \cdots = S[Q[k-1]] = 1$ it is clear that E_k is non-negatively correlated with $\cup_{j=k+1}^{N} A_{m_j}$, thus

$$\mathrm{P}\left(\bigcup_{j=0}^{N} A_j | E_k\right) \geqslant \mathrm{P}\left(\bigcup_{j=k+1}^{N} A_{m_j} | E_k\right) \geqslant \mathrm{P}\left(\bigcup_{j=k+1}^{N} A_{m_j}\right). \tag{4}$$

The inductive hypothesis (applied to the indices $\{i_{m_j}\}$) yields

$$\mathrm{P}\left(\bigcup_{j=k+1}^{N} A_{m_j}\right) \geqslant \mathrm{P}\left(\bigcup_{j=k+1}^{N} \tilde{A}_j\right).$$

The latter inequality combined with (3) and (4) completes the proof of (2) and therefore of (1).

Finally, we have to prove that for $i_j = j$,

$$\mathrm{P}\left(\bigcup_{j=0}^{n-1} A_j\right) > \mathrm{P}\left(\bigcup_{j=0}^{n-1} \tilde{A}_j\right). \tag{5}$$

We prove (5) by induction on n. For $n = 2$ we have

$$\mathrm{P}\left(\bigcup_{j=0}^{1} A_j\right) = 2p^W - p^{2W - |Q \cap (1+Q)|} > 2p^W - p^{W+1} = \mathrm{P}\left(\bigcup_{j=0}^{1} \tilde{A}_j\right).$$

As for the inductive step, note that the proof of (1) shows that for all $k = 0, 1, \ldots, W$,

$$\mathrm{P}\left(\bigcup_{j=0}^{n-1} A_j | E_k\right) \geqslant \mathrm{P}\left(\bigcup_{j=0}^{n-1} \tilde{A}_j | \tilde{E}_k\right).$$

Thus, (5) will be established if we can prove that

$$\mathrm{P}\left(\bigcup_{j=0}^{n-1} A_j | E_0\right) > \mathrm{P}\left(\bigcup_{j=0}^{n-1} \tilde{A}_j | \tilde{E}_0\right).$$

The latter follows from the inductive hypothesis as follows:

$$\mathrm{P}\left(\bigcup_{j=0}^{n-1} A_j | E_0\right) = \mathrm{P}\left(\bigcup_{j=2}^{n-1} A_j\right) > \mathrm{P}\left(\bigcup_{j=2}^{n-1} \tilde{A}_j\right) = \mathrm{P}\left(\bigcup_{j=1}^{n-1} \tilde{A}_j | E_1\right). \qquad \square$$

We next establish an upper bound on $E[\tau_Q]$ which is sharp for the naive seed $\tilde{Q}$. Let $M = s(Q)$ then,

Claim 2.

$$E[\tau_Q] \leqslant \sum_{k=0}^{M-1} p^{-g(k)} - M,$$

where for $k = 0, 1, \ldots, M-1$, $g(k) = |Q_0 \cap Q_k|$, in other words it is the number of 1s that coincide between the seed and a k units shifted version of it.

Remark. Let $\psi(Q) = \sum_{k=0}^{M-1} p^{-g(k)} - M$. Since we know that for the contiguous seed $\tilde{Q}$, $E[\tau_{\tilde{Q}}] = \psi(\tilde{Q})$, we can also prove (the weak version of) Corollary 2 by showing that for all Q, $\psi(Q) \leqslant \psi(\tilde{Q})$. The latter inequality can be established by induction on the weight, W.

Proof of Claim 2. The proof is a variation on a classical betting scheme (e.g. [17, E 10.6]). Consider the following gambling game: a random 0-1 string, S, is generated by Bernoulli trials with $\mathrm{P}(1) = p$. For $n = 0, 1, \ldots$, just before the nth trial, or round, a new gambler comes in and bets \$1 that the next bit will be 1 (or that $S[n+Q[0]] = 1$). If he loses he quits; otherwise, our gambler wins $1/p$ dollars which he then bets on $S[n+Q[1]] = 1$ in round $(n+Q[1])$. If he loses this second bet of his he quits; otherwise, he gets $1/p^2$ dollars which he will bet on $S[n+Q[2]] = 1$ in round $(n+Q[2])$, and so on. When a gambler wins all his W bets he quits with all his winnings. Let $\boldsymbol{X}_n$ be the bank's total winnings after the nth round. Since the game is fair it is not hard to see that $\boldsymbol{X}$ is a martingale. Let ρ_Q be the first time a gambler walks out winning (note that $\rho_Q = \tau_Q + M - 1$). Then it is easy to see that ρ_Q is a stopping time and that if $p > 0$ then $E[\rho_Q] < \infty$. Since $\boldsymbol{X}$ has bounded increments, by Doob's optional stopping theorem (e.g. [17]) $E[\mathbf{X}_{\rho_Q}] = 0$. After the ρ_Qth round the bank received $\rho_Q + 1$ dollars from that many gamblers who joined the game at the various rounds. On the other hand the bank paid Y dollars to the, up to M players, who did not loose yet (clearly $Y \geqslant p^{-W}$, the amount paid to the first lucky winner). Thus,

$$E[\rho_Q] + 1 = E[Y] = \sum_{i=0}^{M-1} E[Y_i],$$

where Y_i is the winnings of the $(\tau_Q + i)$th $(=\rho_Q - (M-1) + i)$ gambler immediately after the ρ_Qth round (for example $Y_0 \equiv p^{-W}$). We finish our proof by showing that

$$E[Y_i] \leqslant p^{-g(i)} \text{ for } i \in \{0, 1, \ldots, M-1\}.$$

For $i \in \{0, 1, \ldots, M-1\}$ define the shape $Q_i = (i + Q) \cap \{0, 1, \ldots, M-1\}$[5] and let $h(i) = |Q_i|$. For example, $h(0) = W$, $h(1) = W - 1$, and $h(2)$ is $W - 2$ or $W - 1$ depending on whether or not $M - 2 \in Q$. Then $Y_i = p^{-h(i)}$ or $Y_i = 0$ depending on whether or not the $(\tau_Q + i)$th gambler won all his $h(i)$ bets, or equivalently on whether or not $S[\tau_Q + Q_i] = \mathbf{1}_{h(i)}$. We know he won $g(i)$ of his (potential) bets, since we know that $S[\tau_Q + Q] = \mathbf{1}_W$ and that $|(\tau_Q + Q) \cap (\tau_Q + Q_i)| = |Q \cap (i + Q)| = g(i)$. Let $Q' = Q_i \setminus Q$; then, $Y_i = p^{-h(i)}$ if and only if $S[\tau_Q + Q'] = \mathbf{1}_{h(i)-g(i)}$. Suppose for a moment that $\mathrm{P}(S[\tau_Q + Q'] = \mathbf{1}_{h(i)-g(i)}) \leqslant p^{(h(i)-g(i))}$, in this case our proof is complete since then

$$E[Y_i] = p^{-h(i)} \cdot \mathrm{P}(S[\tau_Q + Q'] = \mathbf{1}_{h(i)-g(i)}) \leqslant p^{-h(i)} \cdot p^{(h(i)-g(i))} = p^{-g(i)}.$$

Why is $\mathrm{P}(S[\tau_Q + Q'] = \mathbf{1}_{h(i)-g(i)}) \leqslant p^{(h(i)-g(i))}$? Clearly, for any k, $\mathrm{P}(S[k + Q'] = \mathbf{1}_{h(i)-g(i)}) = p^{h(i)-g(i)}$. The reason we have an inequality is that we are given that the τ_Qth gambler was the *first* to win all his bets and conditional on that all other bets prior to ρ_Q (for which we have no direct knowledge) are less likely to be successful. More precisely, since $(k + Q') \cap (k + Q) = \emptyset$ one can easily show that for any vector $\boldsymbol{b}$ of $h(i) - g(i)$ bits,

$$\begin{aligned} &\mathrm{P}(\tau_Q = k | S[k + Q'] = \boldsymbol{b}, S[k + Q] = \mathbf{1}_W) \\ &\qquad \geqslant \mathrm{P}(\tau_Q = k | S[k + Q'] = \mathbf{1}_{h(i)-g(i)}, S[k + Q] = \mathbf{1}_W). \end{aligned}$$

An application of Bayes' law completes our proof

$$\begin{aligned} &\mathrm{P}(S[\tau_Q + Q'] = \mathbf{1}_{h(i)-g(i)} | \tau_Q = k) \\ &\quad = \frac{\mathrm{P}(\tau_Q = k | S[k + Q'] = \mathbf{1}, S[k + Q] = \mathbf{1}_W) \mathrm{P}(S[k + Q'] = \mathbf{1})}{\sum_{\boldsymbol{b}} \mathrm{P}(\tau_Q = k | S[k + Q'] = \boldsymbol{b}, S[k + Q] = \mathbf{1}_W) \mathrm{P}(S[k + Q'] = \boldsymbol{b})} \\ &\quad \leqslant \frac{\mathrm{P}(S[k + Q'] = \mathbf{1})}{\sum_{\boldsymbol{b}} \mathrm{P}(S[k + Q'] = \boldsymbol{b})} \\ &\quad = p^{h(i)-g(i)}. \qquad \square \end{aligned}$$

[5] Strictly speaking Q_i is not a shape since $0 \notin Q_i$ (for $i > 0$). Nevertheless, for our notational purposes we can still consider Q_i as a shape.

4. Finding optimal seeds

We find the optimal seed of weight W and maximal length (span) M by looking for a seed that will maximize the hitting probability among all $\binom{M}{W}$ such seeds. In this section, we describe two methods for computing the hitting probability for a given seed.

4.1. A dynamic programming to compute the exact sensitivity

Let S be a random 0-1 string of length L. Each bit independently is 1 with probability p. Let Q be a seed with weight W, length M. In what follows we will identify the notion of Q as a binary string with the notion of Q as a set of W integers. Let A_i be the event that $S[i+Q]=1_W$, $0 \leqslant i \leqslant L-M$. In this section, we give an algorithm to compute the probability that Q hits S, i.e., $\mathrm{P}(\cup_{j=0}^{L-M} A_j)$.

Let $b=b_0 b_1 \ldots b_{l-1}$ be a binary string of length l.

For any $M \leqslant i \leqslant L$ and any b such that $l=|b| \leqslant M$, we use $f(i,b)$ to denote the probability that Q hits the length i prefix of S that ends with b

$$f(i,b)=\mathrm{P}\left(\bigcup_{j=0}^{i-M} A_j \mid S[i-l,\ldots,i-1]=b\right).$$

Clearly, $\mathrm{P}(\cup_{j=0}^{L-M} A_j)=f(L,\varepsilon)$, where ε denotes the empty string. The idea of our dynamic programming is to compute all $f(i,b)$ gradually for i from M to L, and for all b in a suitably chosen small subset B_1 of $B=\{0,1\}^{\leqslant M}$.

B_1 will contain all b "compatible" with Q, that is all bs for which $A_{i-M} \cap \{S[i-l,\ldots,i-1]=b\} \neq \emptyset$, or equivalently,

$$(\mathbf{1}_{M-l}b)[Q]=\mathbf{1}_W. \tag{6}$$

So b_{l-j} must be 1 whenever $Q[M-j]=1$. The size of B_1 is bounded by $M2^{M-W}$, since for each length $l \leqslant M$, at most $M-W$ bit positions are not constrained by (6).

For $b \in B_0 = B \setminus B_1$, $A_{i-M} \cap \{S[i-l,\ldots,i-1]=b\}=\emptyset$, so in that case

$$f(i,b)=f(i-1,b \gg 1), \tag{7}$$

where $b \gg j$ denotes the binary string $b_0 b_1 \ldots b_{l-1-j}$.

If $b \in B_1$ and $|b|=M$ then $A_{i-M} \supset \{S[i-M,\ldots,i-1]=b\}$, thus

$$f(i,b)=1. \tag{8}$$

In the general case $b \in B_1$ and $|b|<M$ we must consider the bit in S preceding b

$$f(i,b)=(1-p)f(i,0b)+pf(i,1b). \tag{9}$$

Now we are ready to give the DP algorithm to compute all $f(i,b)$ for $M \leqslant i \leqslant L$ and $b \in B_1$.

```
Algorithm DP
Input A seed $Q$, a positive probability $p$, the length $L$ of the region.
Output The probability that $Q$ hits the region.
1. Compute $B_1$;
2. Let $f[i,b]=0$ for all $0 \leqslant i < M$ and $b \in B_1$;
3. for $i$ from $M$ to $L$ do
   for $b$ in $B_1$ from the longest to the shortest do
   if $|b|=M$ then $f[i,b]=1$;
   else
      let $j \geqslant 0$ be the least numbers such that $0b \gg j \in B_1$;
         let $f[i,b]=(1-p)\times f[i-j,0b \gg j]+p\times f[i,1b]$;
4. output $f[L,\varepsilon]$.
```

The correctness of the algorithm follows directly from Formulas (8), (9) and (7). Because $|B_1| < M2^{M-W}$, the algorithm needs $O(M^2 2^{M-W} L)$ time. When $M-W=O(\log L)$, the dynamic programming needs polynomial time.

4.2. Recurrence relationship

In what follows it is convenient to allow shapes (and corresponding Q-grams) with negative offsets. For example, with S as in example 1 and with $Q=\{0,-2,-4\}$, $S[9+Q]=s_5s_7s_9=\mathtt{ATC}$. Let γ be the shape of the seed we are looking for, expressed in negative offsets, and let α and ω be two other shapes where the latter also has negative offsets. α can be thought of as the "prefix" and ω as the "suffix" of S.

Let $Q(L;\alpha,\omega,\gamma)$ be the probability that $S[L+\gamma]=\mathbf{1}_W$ and for all $i<L$, $S[i+\gamma]\neq\mathbf{1}_W$, where S is a random 0-1 string that begins with the all 1 Q-gram $S[\alpha]=\mathbf{1}_{|\alpha|}$ and ends with the all 1 Q-gram $S[L+\omega]=\mathbf{1}_{|\omega|}$.

Let $P(L;\alpha,\omega,\gamma)$ be the probability that for all $i\leqslant L$, $S[i+\gamma]\neq\mathbf{1}_W$, where S is as above. Clearly, the probability we are interested in is $1-P(L;\emptyset,\emptyset,\gamma)$. The following intertwined recursion relation holds between the Ps and the Qs:

$$P(L;\alpha,\omega,\gamma)=1-\sum_{k=s(\gamma)}^{L} Q(k;\alpha,\omega\oplus(L-k),\gamma),$$

where $\omega\oplus i$ is the set $\{i+j : j\in\omega \text{ and } i+j\leqslant 0\}$, intuitively it is the suffix of S after the last i positions of S are chopped. As for Q, clearly if $L<s(\gamma)$, then $Q(L;\alpha,\omega,\gamma)=0$. Otherwise, if $L=s(\gamma)$ then $Q(L;\alpha,\omega,\gamma)=p^{|(L+\gamma)\setminus(\alpha\cup(L+\omega))|}$. Finally, if $L>s(\gamma)$ then $Q(L;\alpha,\omega,\gamma)$ can be found by multiplying the probability that $S[L+\gamma]=\mathbf{1}_W$ by the conditional probability that for $i<L$ $S[i+\gamma]\neq\mathbf{1}_W$ *given* that $S[L+\gamma]=\mathbf{1}_W$. More precisely, for $L>s(\gamma)$ we have

$$Q(L;\alpha,\omega,\gamma)=p^{|(L+\gamma)\setminus(\alpha\cup(L+\omega))|}P(L-1;-[(\omega\cup\gamma)\oplus 1],-\alpha,-[s(\gamma)+\gamma]),$$

where for a shape β, the set $-\beta=\{-i : i\in\beta\}$.

These recurrence relations can be translated to another algorithm which computes the probability of a hit, or $1 - P(L; \emptyset, \emptyset, \gamma)$. As for the complexity, it is easy to see that it is bounded by $Ls(\gamma)$ times the number of different $P(k; \alpha, \omega, \gamma)$ that we need to compute. Clearly $k \leqslant L$ and a look at the recursion equations will show you that the αs and ωs that will come up during the computation are all essentially generated by unions of translations of the shape γ (or $-\gamma$). Thus, the complexity can vary sharply from being polynomial for a shape such as the contiguous one, to being exponential in $s(\gamma) - |\gamma|$ for certain other shapes. In any case, it is bounded by $L^2 s(\gamma) 2^{2(s(\gamma)-|\gamma|)}$. The question of the average complexity is yet to be settled.

5. Conclusions

We present an algorithm for computing the sensitivity of a given seed. In order to find the optimal seed of a given weight, W, and maximal span, M, we simply enumerate over all such possible seeds, applying our aforementioned algorithm to each seed. This works reasonably well in practice: the program which is written in Java finds the optimal seed with $W = 11$ and $M \leqslant 18$ in about 10 min on a 2.8 GHz Pentium 4 PC. We consider this reasonably fast given that this exhaustive search is not likely to be repeated too often: once an optimal seed has been identified it is coded into the Blast-like engine.

On the theoretical side we view our results as first steps toward identifying a set of general conditions under which one can demonstrate that spaced seeds are provably better than the contiguous one.

Acknowledgements

The first author would like to thank Pavel Pevzner for his encouragement and support while working on this project.

References

[1] S.F. Altschul, W. Gish, W. Miller, E. Myers, D.J. Lipman, Basic local alignment search tool, J. Mol. Biol. 215 (1990) 403–410.

[2] S.F. Altschul, T.L. Madden, A.A. Schäffer, J. Zhang, Z. Zhang, W. Miller, D.J. Lipman, Gapped blast and psi-blast: a new generation of protein database search programs, Nucleic Acids Res. 25 (1997) 3389–3402.

[3] J. Buhler, Efficient large-scale sequence comparison by locality-sensitive hashing, Bioinformatics 17 (2001) 419–428.

[4] S. Burkhardt, J. Kärkkäinen, Better filtering with gapped q-grams, In CPM 2001, Lecture notes in computer science, Jerusalem, Israel, Springer 2089 (2001) 73–85.

[5] A. Califano, I. Rigoutsos, FLASH: fast look-up algorithm for string homology. Technical Report, IBM T.J. Watson Research Center, 1995.

[6] W. Gish, WU-Blast 2.0. Website: http://blast.wustl.edu.

[7] X. Huang, W. Miller, A time-efficient, linear-space local similarity algorithm, Adv. Appl. Math. 12 (1991) 337–357.

[8] International human genome sequencing consortium, Initial sequencing and analysis of the human genome, Nature 409 (2001) 860–921.

[9] D.J. Lipman, W.R. Pearson, Rapid and sensitive protein similarity searches, Science 227 (1985) 1435–1441.

[10] B. Ma, J. Tromp, M. Li, Patternhunter—faster and more sensitive homology search, Bioinformatics 18 (3) (2002) 440–445.

[11] F.P. Preparata, A.M. Frieze, E. Upfal, On the power of universal bases in sequencing by hybridization. In S. Istrail, P.A. Pevzner, M.S. Waterman, (Eds.), Proceedings of the Third Annual International Conference on Computational Molecular Biology (RECOMB-99), pages 295–301, Lyon, France, April 1999. ACM Press.

[12] F.P. Preparata, E. Upfal, Sequencing-by-hybridization at the information-theory bound: an optimal algorithm. In R. Shamir, S. Miyano, S. Istrail, P.A. Pevzner, M.S. Waterman, (Eds.), Proceedings of the Fourth Annual International Conference on Computational Molecular Biology (RECOMB-00), pages 245–253, Tokyo, Japan, April 2000. ACM Press.

[13] S.M. Ross, Stochastic Processes, Wiley Series in Probability and Statistics: Probability and Statistics, 2nd Edition, Wiley, New York, 1996.

[14] D. States, SENSEI website: http://stateslab.wustl.edu/software/sensei/

[15] T.A. Tatusova, T.L. Madden, Blast 2 sequences—a new tool for comparing protein and nucleotide sequences, FEMS Microbiol. Lett. 174 (1999) 247–250.

[16] J.C. Venter, et al., The sequence of the human genome, Science 291 (2001) 1304.

[17] D. Williams, Probability with Martingales, Cambridge University Press, Cambridge, 1991.

[18] Z. Zhang, S. Schwartz, L. Wagner, W. Miller, A greedy algorithm for aligning DNA sequences, J. Comput. Biol. 7 (1-2) (2000) 203–214.