Bounds for Resequencing By Hybridization

Dekel Tsur*

Abstract

We study the problem of finding the sequence of an unknown DNA fragment given the set of its k-long subsequences and a homologous sequence, namely a sequence that is similar to the target sequence. Such a sequence is available in some applications, e.g., when detecting single nucleotide polymorphisms. Pe'er and Shamir studied this problem and presented a heuristic algorithm for it. In this paper, we give an algorithm with provable performance: We show that under some assumptions, the algorithm can reconstruct a random sequence of length $O(4^k)$ with high probability. We also show that no algorithm can reconstruct sequences of length $\Omega(\log k \cdot 4^k)$.

1 Introduction

Sequencing by Hybridization (SBH) [3,22] is a method for sequencing of long DNA molecules. Using a chip containing all 4^k sequences of length k one can obtain the set of all k-long subsequences of the target sequence: For every sequence in the chip, if its reverse complement appears in the target than the two sequences will hybridize. The set of all k-long subsequences of the target is called the *k-spectrum* (or *spectrum*) of the target. After obtaining the spectrum, the target sequence can be reconstructed in polynomial time [26].

Unfortunately, other sequences can have the same spectrum as the target's. For example, if we assume that the sequence is chosen uniformly from all the sequences of length n, then only sequences of length less than roughly 2^k can be reconstructed reliably [2, 13, 27, 29]. Several methods for overcoming this limitation of SBH were proposed: gapped probes [14, 16, 18–20, 27, 28], interactive protocols [15, 23, 31, 34], using location information [1, 4, 11, 12, 17, 29], and using restriction enzymes [30, 32].

An additional limitation of SBH is that in practice, there are errors in the hybridization process. Thus, some subsequences of the target do not appear in the experimental spectrum (*false negatives*), and the experimental spectrum

*School of Computer Science, Tel Aviv University. E-Mail: dekelts@tau.ac.il

contains sequences that do not appear in the target (*false positives*). Several algorithms were given for SBH with errors [5–10, 21, 26]. The first algorithm with provable performance was given by Halperin et al. [16], and their algorithm was later improved in [33].

In many applications, the target sequence is not completely unknown. For example, the problem of detecting single nucleotide polymorphisms can be considered as finding the sequence of a DNA fragment when most of the sequence (over 99%) is known in advance. Therefore, we study the problem of reconstructing a target sequence given its spectrum and a sequence that is similar to the target sequence (called a *homologous sequence*). This problem is called *resequencing by hybridization (RBH)*. Pe'er and Shamir [25] (see also [24]) gave an algorithm for RBH and showed that the algorithm works well in practice, but did not prove a bound on its performance.

In this work, we give an algorithm for RBH and prove a bound on its performance. We assume that the target sequence is a random sequence, and that the homologous sequence has the following property: Every k-subsequence of the target differs from the corresponding subsequence of the homologous sequence in at most d letter, where $d \leq (\frac{3}{4} - \delta)k$ for an arbitrarily small constant δ. Moreover, we assume that the homologous sequence is generated randomly from the target sequence by selecting positions on the target sequence and then randomly changing the letters in the selected positions. Under these assumptions, our algorithm can reconstruct sequences of length $O(4^k \min(d^{-3/2}, \log k/k))$ with probability close to 1. We also show that that the algorithm can reconstruct sequences of length $O(4^k)$ if the number of different letters between the target and the homologous sequence is $O(n/2^{\epsilon k})$. Moreover, we show that no algorithm can reconstruct sequences of length $\Omega(\log k \cdot 4^k)$ with success probability greater than $\frac{1}{3}$. We also study the RBH problem under the presence of hybridization errors. We give an algorithm for this case whose performance is close to the performance of the algorithm in the errorless case.

Due to lack of space, some proofs are omitted.

2 Preliminaries

For a sequence $A = a_1 \cdots a_n$, let A_i^l denote the l-subsequence $a_i a_{i+1} \cdots a_{i+l-1}$. For two sequences A and B, AB is the concatenation of A and B.

A set $I \subseteq \{1, \ldots, n\}$ is called a k, d-*set* if $|I \cap \{i, \ldots, i+k-1\}| \leq d$ for all i. We denote by $I_{k,d}(n)$ the set of all k, d-sets that are subsets of $\{1, \ldots, n\}$. Two sequences $A = a_1 \cdots a_n$ and $B = b_1 \cdots b_n$ will be called k, d-*equal* if $\{i : a_i \neq b_i\}$ is a k, d-set.

The RBH problem is as follows: Given the k-spectrum of A and a sequence H which is k, d-equal to A, find the sequence A. We study the RBH problem under a probabilistic model. We assume that the target sequence $A = a_1 \cdots a_n$ is

1. Let $s_1, s_2, \ldots, s_{k-1}$ be the first $k-1$ letters of A.
2. Let $s_{n-k+2}, \ldots, s_n$ be the last $k-1$ letters of A.
3. For $t = k, k+1, \ldots, \lceil n/2 \rceil$ do: (forward sequencing)
   (a) Let $\mathcal{B}$ be the set of all sequences of length ck that are k, d-equal to H_t^{ck}.
   (b) Let $\mathcal{B}'$ be the set of all sequences $B \in \mathcal{B}$ such that all the k-subsequences of $S_{t-k+1}^{k-1}B$ appear in A.
   (c) If all the sequences in $\mathcal{B}'$ have a common first letter a, then set $s_t \leftarrow a$. Otherwise, set $s_t \leftarrow h_t$.
4. For $t = n-k+1, n-k, \ldots, \lceil n/2 \rceil + 1$ do: (backward sequencing)
   (a) Let $\mathcal{B}$ be the set of all sequences of length ck that are k, d-equal to H_{t-ck+1}^{ck}.
   (b) Let $\mathcal{B}'$ be the set of all sequences $B \in \mathcal{B}$ such that all the k-subsequences of BS_{t+1}^{k-1} appear in A.
   (c) If all the sequences in $\mathcal{B}'$ have a common last letter a, then set $s_t \leftarrow a$. Otherwise, set $s_t \leftarrow h_t$.
5. Return the sequence S.

Figure 1: Algorithm A.

chosen at random, where each letter is chosen uniformly from $\Sigma = \{A, C, G, T\}$ and independently of the other letters. The homologous sequence $H = h_1 \cdots h_n$ is built as follows: Some k, d-set I (called the *locations set*) is chosen before the sequence A is chosen. After the sequence A is chosen, for every $i \notin I$ set $h_i = a_i$, and for every $i \in I$, h_i is chosen uniformly from $\Sigma - \{a_i\}$.

We will use $\log x$ to denote the logarithm with base 2 of x.

3 Algorithm for RBH

In this section we give an algorithm for solving RBH, and analyze its performance. We assume that the first and last $k-1$ letters of A are known. Let c be some large integer, and suppose that n is large enough so $ck \leq n/2$. We will use $S = s_1 \cdots s_n$ to denote the sequence that is built by our algorithm. The algorithm is given in Figure 1.

As the backward sequencing stage is analogous to the forward sequencing

stage, we shall only analyze the latter. A sequence in the set $\mathcal{B}$ in some step t of the algorithm is called a *path (w.r.t. t)*. A path is called *correct* if it is equal to $a_t \cdots a_{t+ck-1}$, and it is called *incorrect* if its first letter is not equal to a_t. An incorrect path in $\mathcal{B}'$ is called a *bad path*. For a path $B \in \mathcal{B}$, a *supporting probe* is a k-subsequence of B which is also a subsequence of the target sequence A. We will use i to denote the supporting probe B_i^k.

Clearly for every t, the correct path is always in $\mathcal{B}'$. Thus, if $a_t = h_t$ then the algorithm will always set s_t to a_t. In other words, the algorithm can fail only at indices t for which $a_t \neq h_t$.

We first give two technical lemmas:

Lemma 1. *For every $\alpha \geq 0$, if $d \leq \left(\frac{3}{4} - \sqrt{\frac{1}{c} + (\frac{1}{2}\log e)\alpha + \frac{\log k}{k}}\right) k$ then $k^2\binom{k}{d} \cdot 3^d e^{\alpha k}/4^{(1-1/c)k} \leq \frac{1}{2}$.*

Proof. Let $a = d/k$. Using Stirling formula we have that

$$\begin{aligned}\binom{k}{d} &\leq \frac{1.1\sqrt{2\pi k}\left(\frac{k}{e}\right)^k}{\sqrt{2\pi d}\left(\frac{d}{e}\right)^d \sqrt{2\pi(k-d)}\left(\frac{k-d}{e}\right)^{k-d}} \\ &= \frac{1.1}{\sqrt{2\pi}}\sqrt{\frac{k}{d(k-d)}} \cdot \frac{k^k}{(ak)^{ak}((1-a)k)^{(1-a)k}} \\ &< \frac{1}{2} \cdot \frac{1}{a^{ak}(1-a)^{(1-a)k}} = \frac{1}{2} \cdot 2^{\left(a\log\frac{1}{a} + (1-a)\log\frac{1}{1-a}\right)k}.\end{aligned}$$

Thus, $\log(2\binom{k}{d} \cdot 3^d) \leq (a\log\frac{1}{a} + (1-a)\log\frac{1}{1-a} + a\log 3)k$.

Let $f(x) = x\log\frac{1}{x} + (1-x)\log\frac{1}{1-x} + x\log 3$. Using simple calculus, we obtain that $f(\frac{3}{4} - x) \leq 2(1-x^2)$. Therefore,

$$\begin{aligned}\log\left(2\binom{k}{d} \cdot 3^d\right) &\leq f(a)k \leq 2(1-(3/4-a)^2)k \\ &\leq 2\left(1 - \frac{1}{c} - \frac{\log e}{2}\alpha - \frac{\log k}{k}\right)k = \log\left(\frac{4^{(1-1/c)k}}{e^{\alpha k}k^2}\right).\end{aligned}$$

■

Lemma 2. *Let $F = \sum_{j=1}^{k} \frac{(a+j)e^{\alpha j}}{4^j}\sum_{i=0}^{\min(j,d)}\binom{j}{i}3^i$ and $F' = \sum_{j=1}^{k}\frac{e^{\alpha j}}{4^j}\cdot\sum_{i=0}^{\min(j,d)}\binom{j}{i}3^i$. If $a \geq 0$ and $0 \leq \alpha \leq 0.14$ then $F \leq (2d+6)(a+2d)e^{2\alpha d}$ and $F' \leq (2d+3)e^{2\alpha d}$.*

Proof. Let $f(j) = \sum_{i=0}^{\min(j,d)}\binom{j}{i}3^i$. For every j,

$$f(j) \leq \sum_{i=0}^{j}\binom{j}{i}3^i = 4^j.$$

If $j > 2d$ then

$$f(j) = \sum_{i=0}^{d} \binom{j}{i} 3^i \le \binom{j}{d} \sum_{i=0}^{d} 3^i \le \frac{3}{2}\binom{j}{d} 3^d.$$

Therefore,

$$F \le \sum_{j=1}^{2d} (a+j)e^{\alpha j} + \sum_{j=2d+1}^{k} (a+j)\left(\frac{e^\alpha}{4}\right)^j \cdot \frac{3}{2}\binom{j}{d} 3^d,$$

where the second sum is empty if $d > (k-1)/2$. We have that

$$\sum_{j=1}^{2d} (a+j)e^{\alpha j} \le \sum_{j=1}^{2d} (a+2d)e^{\alpha \cdot 2d} = 2d(a+2d)e^{2\alpha d},$$

Furthermore, for $j > 2d$,

$$\frac{\binom{j}{d}\left(\frac{e^\alpha}{4}\right)^j}{\binom{j-1}{d}\left(\frac{e^\alpha}{4}\right)^{j-1}} = \frac{j!}{d!(j-d)!} \cdot \frac{d!(j-1-d)!}{j-1!} \cdot \frac{e^\alpha}{4} = \frac{j}{j-d} \cdot \frac{e^\alpha}{4} \le 2 \cdot \frac{e^{0.14}}{4} \le \frac{3}{5},$$

so

$$\binom{j}{d}\left(\frac{e^\alpha}{4}\right)^j \le \binom{2d}{d}\left(\frac{e^\alpha}{4}\right)^{2d}\left(\frac{3}{5}\right)^{j-2d} \le 2^{2d}\left(\frac{e^{0.14}}{4}\right)^{2d}\left(\frac{3}{5}\right)^{j-2d}.$$

Thus,

$$\begin{aligned}
\sum_{j=2d+1}^{k} (a+j)\left(\frac{e^\alpha}{4}\right)^j \cdot \frac{3}{2}\binom{j}{d} 3^d &\le \frac{3}{2} \cdot 3^d \left(\frac{e^{0.14}}{2}\right)^{2d} \sum_{l=1}^{\infty} (a+2d+l)\left(\frac{3}{5}\right)^l \\
&= \frac{3}{2} \cdot \left(\frac{e^{0.14}\sqrt{3}}{2}\right)^{2d} \left(\frac{3}{2}(a+2d) + \frac{15}{4}\right) \\
&< \frac{3}{2}\left(\frac{3}{2}(a+2d) + \frac{15}{4}\right) < 6(a+2d),
\end{aligned}$$

so the bound on F follows. The bound on F' is proved using similar analysis. ■

Theorem 3. *For every $\epsilon > 0$, if $d \le \left(\frac{3}{4} - \sqrt{1/c + O(\log k/k)}\right)k$ and $n = O(\epsilon 4^k \min(d^{-3/2}, \log k/k))$ then the probability that algorithm A fails is at most ϵ.*

Proof. Let E_t be the event that t is the minimum index for which $s_t \neq a_t$. Assuming that events $E_1, \ldots, E_{t-1}$ do not happen, E_t happens if and only if $a_t \neq h_t$ and there is a bad path w.r.t. t. To bound this probability for some fixed t, we fix a k, d-set $I \subseteq \{1, \ldots, ck\}$ and generate a random path $B' = b'_1 \cdots b'_{ck}$ as follows: If $i \notin I$, then $b'_i = h_{t-1+i}$. If $i > 1$ and $i \in I$ then b'_i is selected

uniformly at random from $\Sigma - \{h_{t-1+i}\}$, and if $i = 1$ and $i \in I$ then b'_i is selected uniformly from $\Sigma - \{h_t, a_t\}$. We shall compute the probability that B' is a bad path w.r.t. t, and then we will bound $\mathrm{P}[E_t]$ by roughly $\sum_I 3^{|I|} \cdot \mathrm{P}[B' \text{ is bad}|t, I]$, where the term $3^{|I|}$ bounds the number of possible paths B' given the set I. Note that each letter of B' has a uniform distribution over Σ, and the letters of B' are independent.

Let $B = S^{k-1}_{t-k+1}B'$, and denote $B = b_1 \cdots b_{ck+k-1}$. By definition, B' is a bad path if and only if there are indices $r_1, \ldots, r_{ck}$ such that $B^k_i = A^k_{r_i}$ for $i = 1, \ldots, ck$, so we need to bound the probability that these events happen. We say that supporting probe i is *trivial* if $r_i = t - 1 + i$. Note that probes $1, \ldots, k$ are not trivial as $b'_1 \neq a_t$. We consider two cases: The first case is when there are no trivial supporting probes, and the second case is when there are trivial supporting probes. These cases will be called case I and case II, respectively.

Case I Suppose that there are no trivial supporting probes. The difficulty in bounding the probability that $B^k_i = A^k_{r_i}$ for $i = 1, \ldots, ck$ is that these events are not independent when some of the sequences $A^k_{r_1}, \ldots, A^k_{r_k}$ have common letters, that is, if $|r_i - r_j| < k$ for some pairs (i, j) of indices. We say that two probes r_i and r_j are *strongly adjacent* if $|r_i - r_j| < k$ and $r_j - r_i = j - i$ (in particular, every probe is strongly adjacent to itself). The transitive closure of the strongly adjacency relation will be called the *adjacency* relation. The motivation behind the definitions above is as follows: If r_i and r_j are strongly adjacent probes with $i < j$, then the events $B^k_i = A^k_{r_i}$ and $B^k_j = A^k_{r_j}$ happen if and only if $B^{k+j-i}_i = A^{k+j-i}_{r_i}$. More generally, for each equivalence class of the adjacency relation, there is a corresponding equality event between a subsequence of A and a subsequence of B.

If r_i and $r_{i'}$ are adjacent, then $B^k_j = A^k_{r_i+j-i}$ for every $j = i, \ldots, i'$. Therefore, we can assume w.l.o.g. that $r_j = r_i + j - i$ for $j = i, \ldots, i'$. Thus, each equivalence class of the adjacency relation corresponds to an interval in $\{1, \ldots, ck\}$. More precisely, there are indices $1 = c_1 < c_2 < \cdots < c_x < c_{x+1} = ck + 1$ such that $\{r_{c_i}, r_{c_i} + 1, \ldots, r_{c_{i+1}}\}$ is an equivalence class for $i = 1, \ldots, x$. The sequence B' is a bad path if and only if $B^{k-1+c_{i+1}-c_i}_{c_i} = A^{k-1+c_{i+1}-c_i}_{r_{c_i}}$ for all i. We shall compute the probability that these events happen. Each sequence $A^{k-1+c_{i+1}-c_i}_{r_{c_i}}$ will be called a *block*, and will be denoted by L_i. We denote by $l_i = k - 1 + c_{i+1} - c_i$ the number of letters in the block L_i. To simplify the presentation, we define block L_0 to be the sequence $A^{(c+1)k-1}_{t-k+1}$ ($l_0 = (c + 1)k - 1$).

For two blocks L_i and L_j with $0 \leq i < j$ we say that L_j *overlaps with* L_i if the two blocks have common letters, namely if $r_{c_j} \in [r_{c_i} - l_j + 1, r_{c_i} + l_i - 1]$. If the block L_j overlaps with some block L_i, we say that L_j is an *overlapping block*.

We will bound the probability that B' is a bad path in two cases: When there are no overlapping probes, and when there are overlapping probes. In the second case, we will look at the overlapping probe with minimum index, and consider

the equality events that correspond to this block and the blocks with smaller indices. However, the analysis of this case can be complicated if the overlapping probe overlaps with two or more blocks. Therefore, we introduce the following definition: A block L_j is called *weakly overlapping* if there is an index $i < j$ such that $|r_{c_j} - r_{c_i}| \leq 2(c+1)k - 4$. In particular, an overlapping block is also a weakly overlapping block. If there are overlapping blocks, define y to be the minimum index such that block L_y is a weakly overlapping block. The definition of a weakly close block ensures that L_y cannot overlap with more than one block.

We consider three cases:

1. There are no overlapping blocks.
2. There are overlapping blocks and $y > 1$.
3. There are overlapping blocks and $y = 1$.

Let $\mathcal{E}_i$ denote the event that there is a bad path that satisfies case i above. We shall bound the probability of each of these events.

Case 1 Suppose that there are no overlapping blocks. In this case, the events $B_{c_1}^{l_1} = A_{r_{c_1}}^{l_1}, \ldots, B_{c_x}^{l_x} = A_{r_{c_x}}^{l_x}$ are independent. Therefore, for fixed t, I, and $r_1, \ldots, r_k$, the probability that event $\mathcal{E}_1$ happens is $\prod_{i=1}^{x} 4^{-l_i} = 4^{-ck-(k-1)x}$. For fixed x, the number of ways to choose $c_1, \ldots, c_x$ is $\binom{ck-1}{x-1}$, and for fixed $c_1, \ldots, c_x$, the number of ways to choose $r_1, \ldots, r_k$ is at most n^x. Thus,

$$\begin{aligned}
\mathrm{P}\left[\mathcal{E}_1 | t, I\right] &\leq \sum_{x=1}^{ck} \binom{ck-1}{x-1} \frac{n^x}{4^{ck+(k-1)x}} = \frac{n}{4^{ck+k-1}} \sum_{x=1}^{ck} \binom{ck-1}{x-1} \left(\frac{n}{4^{k-1}}\right)^{x-1} \\
&= \frac{n}{4^{(c+1)k-1}} \left(1 + \frac{n}{4^{k-1}}\right)^{ck-1} \leq \frac{n}{4^{(c+1)k-1}} \cdot e^{(n/4^{k-1}) \cdot ck}.
\end{aligned}$$

Therefore,

$$\mathrm{P}\left[\mathcal{E}_1 | t\right] \leq \sum_{I \in I_{k,d}(ck)} 3^{|I|} \cdot \frac{n e^{(n/4^{k-1})ck}}{4^{(c+1)k-1}}.$$

Let $J_i = \{ik+1, \ldots, ik+k\}$. From the definition of k, d-set we get that

$$\begin{aligned}\sum_{I\in I_{k,d}(ck)} 3^{|I|} &= \sum_{I\in I_{k,d}(ck)} \prod_{i=1}^{c} 3^{|I\cap J_i|} \\ &= \sum_{j_1,\ldots,j_c=0}^{d} |\{I \in I_{k,d}(ck) : |I\cap J_i| = j_i, i=1,\ldots,c\}| \cdot \prod_{i=1}^{c} 3^{j_i} \\ &\le \sum_{j_1,\ldots,j_c=0}^{d} \prod_{i=1}^{c} \binom{k}{j_i} \cdot \prod_{i=1}^{c} 3^{j_i} = \left(\sum_{j=0}^{d} \binom{k}{j} 3^j\right)^c \\ &\le \left((d+1)\binom{k}{d}\cdot 3^d\right)^c,\end{aligned}$$

where the last inequality follows from the fact that $d \le \frac{3}{4}k$.

Since H is k,d-equal to A, it follows that the number of indices i for which $a_i \neq h_i$ is at most $\lceil n/k\rceil d \le 2dn/k$, so the number of ways to choose t is at most $2dn/k$. Therefore, the probability that event $\mathcal{E}_1$ happens is at most

$$\frac{2dn}{k}\cdot\left(k\binom{k}{d}3^d\right)^c \cdot \frac{ne^{(n/4^{k-1})ck}}{4^{(c+1)k-1}} = \frac{8dn^2}{k\cdot 4^{2k}}\cdot\left(\frac{k\binom{k}{d}3^d\cdot e^{(n/4^{k-1})k}}{4^{(1-1/c)k}}\right)^c < \frac{8n^2}{4^{2k}},$$

where the last inequality follows from Lemma 1.

Case 2 Recall that y is the minimum index such that block L_y is a weakly overlapping block. Let $z = c_y$. Let $\mathcal{E}$ be the event that the equality events corresponding to the blocks $L_1,\ldots,L_{y-1}$ happen, namely $B^{l_i}_{c_i} = A^{l_i}_{r_{c_i}}$ for $i = 1,\ldots,y-1$, and let $\mathcal{E}'$ be the event that $B^k_z = A^k_{r_z}$. As there are no overlapping blocks in $L_1,\ldots,L_{y-1}$, we obtain that

$$\mathrm{P}\left[\mathcal{E}|t,I,z,r_1,\ldots,r_{z-1}\right] = \frac{1}{4^{(k-1)(y-1)+z-1}}.$$

Furthermore, $\mathrm{P}\left[\mathcal{E}'|t,I,z,r_z\right] = 4^{-k}$. This is clear when L_y does not overlap with L_0. To see that this claim is also true when L_y overlaps with L_0, note that event $\mathcal{E}'$ is composed of k equalities $b_{z+i} = a_{r_z+i}$ for $i = 0,\ldots,k-1$. The probability that such an equality happens given that the previous equalities happen is exactly $1/4$ as the letters b_{z+i} and a_{r_z+i} are independent (since probe z is not trivial), and at least one of these two letters is not restricted by the the previous equalities. Therefore, $\mathrm{P}\left[\mathcal{E}'|t,I,z,r_z\right] = 4^{-k}$.

Now, we claim that the events $\mathcal{E}$ and $\mathcal{E}'$ are independent. If L_y does not overlap with L_0 then this claim follows from [2, p. 437]. Otherwise, suppose that L_y overlaps with L_0. For each equality $b_{i+i'} = a_{r_{c_i}+i'}$ (where $i = 1,\ldots,y-1$ and $i' = 0,\ldots,l_i-1$) that is induced by $\mathcal{E}$, the letter $a_{r_{c_i}+i'}$ is not restricted by

event $\mathcal{E}'$ (as L_y does not overlap with L_i), and therefore the probability that this equality happen is 1/4. It follows that $\mathcal{E}$ and $\mathcal{E}'$ are independent.

Combining the claims above, we have that

$$\mathrm{P}\left[\mathcal{E} \wedge \mathcal{E}' | t, I, z, r_1, \ldots, r_z\right] = \frac{1}{4^{(k-1)(y-1)+z-1+k}}.$$

For fixed y and z, the number of ways to choose $r_1, \ldots, r_{z-1}$ is at most $\binom{(z-1)-1}{(y-1)-1} n^{y-1}$, and the number of ways to choose r_z is at most $(2(2(c+1)k-4)+1) \cdot (y-1) \leq 4(c+1)kz$. Thus,

$$\begin{aligned}\mathrm{P}\left[\mathcal{E} \wedge \mathcal{E}' | t, I, z\right] &\leq \sum_{y=2}^{z} \binom{(z-1)-1}{(y-1)-1} n^{y-1} \cdot 4(c+1)kz \cdot \frac{1}{4^{(k-1)(y-1)+z-1+k}} \\ &\leq \frac{64(c+1)kzn}{4^{2k+z}} \cdot \left(1 + \frac{n}{4^{k-1}}\right)^{z-2} \leq \frac{64(c+1)kzne^{(n/4^{k-1})z}}{4^{2k+z}}.\end{aligned}$$

The event $\mathcal{E} \wedge \mathcal{E}'$ depends only on the first z letters of B'. Therefore,

$$\mathrm{P}\left[\mathcal{E} \wedge \mathcal{E}'\right] \leq \frac{2dn}{k} \sum_{z=2}^{ck} \sum_{I \in I_{k,d}(z)} 3^{|I|} \cdot \frac{64(c+1)kzne^{(n/4^{k-1})z}}{4^{2k+z}}.$$

Let $z = kz_1 + z_2$ where $1 \leq z_2 \leq k$. Then,

$$\begin{aligned}\sum_{I \in I_{k,d}(z)} 3^{|I|} &= \sum_{I \in I_{k,d}(z)} \prod_{i=1}^{z_1+1} 3^{|I \cap J_i|} \leq \left(\sum_{i=0}^{d} \binom{k}{i} 3^i\right)^{z_1} \sum_{i=0}^{\min(d,z_2)} \binom{z_2}{i} 3^i \\ &\leq \left(k\binom{k}{d} 3^d\right)^{z_1} \sum_{i=0}^{\min(d,z_2)} \binom{z_2}{i} 3^i.\end{aligned}$$

Using Lemma 1 and Lemma 2 we obtain that

$$\begin{aligned}\mathrm{P}\left[\mathcal{E}_2\right] &\leq \mathrm{P}\left[\mathcal{E} \wedge \mathcal{E}'\right] \\ &\leq \frac{128(c+1)dn^2}{4^{2k}} \sum_{z_1=0}^{c-1} \sum_{z_2=1}^{k} \left(k\binom{k}{d} 3^d\right)^{z_1} \\ &\qquad \cdot \sum_{i=0}^{\min(d,z_2)} \binom{z_2}{i} 3^i \cdot \frac{(kz_1+z_2)e^{(n/4^{k-1})(kz_1+z_2)}}{4^{kz_1+z_2}} \\ &\leq \frac{128(c+1)dn^2}{4^{2k}} \sum_{z_1=0}^{c-1} \left(\frac{k\binom{k}{d} 3^d e^{(n/4^{k-1})k}}{4^k}\right)^{z_1} (2d+6)(kz_1+2d)e^{2(n/4^{k-1})d} \\ &\leq \frac{128(c+1)d(2d+6)e^{2(n/4^{k-1})d}n^2}{4^{2k}} \sum_{z_1=0}^{c-1} \left(\frac{k^2\binom{k}{d} 3^d e^{(n/4^{k-1})k}}{4^k}\right)^{z_1} (z_1+2d)\end{aligned}$$

$$\leq \frac{128(c+1)d(2d+6)e^{2(n/4^{k-1})d}n^2}{4^{2k}} \sum_{z_1=0}^{c-1} \frac{z_1+2d}{2^{z_1}}$$
$$\leq \frac{128(c+1)d(2d+6)e^{2(n/4^{k-1})d}n^2}{4^{2k}} \cdot (2+4d).$$

Case 3 For fixed t, I, and r_1, the probability that $B_1^k = A_{r_1}^k$ is 4^{-k}. We multiply this probability by the number of ways to choose t, the number or ways to choose r_1, and by the number of ways to choose b'_1. Thus,

$$\mathrm{P}\left[\mathcal{E}_3\right] \leq \frac{2dn}{k} \cdot 4(c+1)k \cdot 3 \cdot \frac{1}{4^k} = \frac{24(c+1)dn}{4^k}.$$

Case II We now consider the case when there are trivial supporting probes. Let z be the minimum index such that probe $z+k$ is trivial ($z \geq 1$ as probes $1, \ldots, k$ are not trivial). We will consider only the first $z+k-1$ probes. If there are overlapping blocks among the first $z+k-1$ probes, then event $\mathcal{E}_2$ or $\mathcal{E}_3$ happens, so we only need to consider the case when there are no overlapping blocks among these probes. We denote this event by $\mathcal{E}_4$.

Let x be the number of blocks among probes $1, \ldots, z+k-1$. Then,

$$\mathrm{P}\left[\mathcal{E}_4 | t, I, z\right] \leq \sum_{x=1}^{z+k-1} \binom{(z+k-1)-1}{x-1} \frac{n^x}{4^{(k-1)x+z+k-1}} \leq \frac{16ne^{(n/4^{k-1})(z+k)}}{4^{z+2k}}.$$

Since probe $z+k$ is trivial, we have that $b'_{z+1} = a_{t+z}, \ldots, b'_{z+k} = a_{t+z+k-1}$, so event $\mathcal{E}_4$ depends only on the first z letters of B'. Therefore, by Lemma 2,

$$\mathrm{P}\left[\mathcal{E}_4\right] \leq \frac{2dn}{k} \sum_{z=1}^{(c-1)k} \sum_{I \in I_{k,d}(z)} 3^{|I|} \cdot \frac{16ne^{(n/4^{k-1})(z+k)}}{4^{z+2k}}$$
$$\leq \frac{32dn^2e^{(n/4^{k-1})k}}{k \cdot 4^{2k}} \sum_{z_1=0}^{c-2} \sum_{z_2=1}^{k} \left(k\binom{k}{d}3^d\right)^{z_1} \sum_{i=0}^{\min(d,z_2)} \binom{z_2}{i} 3^i \cdot \frac{e^{(n/4^{k-1})(kz_1+z_2)}}{4^{kz_1+z_2}}$$
$$\leq \frac{32dn^2e^{(n/4^{k-1})k}}{k \cdot 4^{2k}} \sum_{z_1=0}^{c-2} \left(\frac{k\binom{k}{d}3^d e^{(n/4^{k-1})k}}{4^k}\right)^{z_1} (2d+3)e^{2(n/4^{k-1})d}$$
$$\leq \frac{32d(2d+3)e^{(n/4^{k-1})(k+2d)}n^2}{k \cdot 4^{2k}} \sum_{z_1=0}^{c-2} \frac{1}{2^{z_1}}$$
$$\leq \frac{32d(2d+3)e^{(n/4^{k-1})(k+2d)}n^2}{k \cdot 4^{2k}} \cdot 2.$$

Combining all four cases, we have that the probability that the algorithm fails is $O((d+e^{(n/4^{k-1})k}/k)d^2e^{2(n/4^{k-1})d}n^2/4^{2k}+dn/4^k)$, so if $n = O(\epsilon 4^k \min(d^{-3/2}, \log k/k))$ then this probability is at most ϵ. ∎

In some applications, the number of different letters between A and H is much smaller than $\lceil n/k \rceil d$, and in that case, the algorithm performs better:

Theorem 4. *For every $\epsilon > 0$, if $d \leq \left(\frac{3}{4} - \sqrt{1/c + O(\log k/k)}\right) k$, $n = O(\epsilon 4^k)$, and the number of different letters between A and H is $O(n/2^{\epsilon k})$, then the probability that algorithm A fails is at most ϵ.*

4 Upper bound

In this section, we show an upper bound on the length of the sequences that can be reconstructed from their spectra and homologous sequences.

We use the following lemma:

Lemma 5. *For every sequence P of length k, the probability that P does not appear in the spectrum of a random sequence of length $n \geq 2(k+1)k$ is at most $e^{-\frac{1}{3}n/4^k}$.*

Proof. Let S be a random sequence of length n, and let A_i denote the event that $S_i^k \neq P$. Clearly, the probability that P does not appear in the spectrum of S is $\mathrm{P}\left[\bigwedge_{i=1}^{n-k+1} A_i\right]$. The difficulty in bounding this probability lies in the fact that the events $A_1, \ldots, A_{n-k+1}$ are dependent. Thus, we will split these events into groups such that the events from one group are independent of the events from other groups.

For $i = 1, \ldots, \lfloor n/2k \rfloor$, let $I_i = \{2k(i-1)+1, \ldots, 2k(i-1)+k+1\}$. Let $B_i = \bigwedge_{j \in I_i} A_j$. The events $B_1, \ldots, B_{\lfloor n/2k \rfloor}$ are independent and have equal probabilities, so

$$\mathrm{P}\left[\bigwedge_{i=1}^{n-k+1} A_i\right] \leq \mathrm{P}\left[\bigwedge_{i=1}^{n-k+1} B_i\right] = \prod_{i=1}^{\lfloor n/2k \rfloor} \mathrm{P}\left[B_i\right] = \mathrm{P}\left[B_1\right]^{\lfloor n/2k \rfloor}.$$

We will now bound $\mathrm{P}\left[B_1\right]$. For an index $i \in I_1$, let $C_i = \overline{A_i} \wedge \bigwedge_{j=1}^{i-1} A_j$. The events $\{C_i\}_{i \in I_1}$ are disjoint, so

$$\mathrm{P}\left[B_1\right] = 1 - \mathrm{P}\left[\bigvee_{i \in I_1} \overline{A_i}\right] \leq 1 - \mathrm{P}\left[\bigvee_{i \in I_1} C_i\right] = 1 - \sum_{i \in I_1} \mathrm{P}\left[C_i\right].$$

For every $i \in I_1$,

$$\begin{aligned}
\mathrm{P}\left[C_i\right] &= \mathrm{P}\left[\,\overline{A_i}\,\right] \cdot \mathrm{P}\left[\bigwedge_{j=1}^{i-1} A_j \,\middle|\, \overline{A_i}\right] = \mathrm{P}\left[\,\overline{A_i}\,\right] \cdot \left(1 - \mathrm{P}\left[\bigvee_{j=1}^{i-1} \overline{A_j} \,\middle|\, \overline{A_i}\right]\right) \\
&\geq \mathrm{P}\left[\,\overline{A_i}\,\right] \cdot \left(1 - \sum_{j=1}^{i-1} \mathrm{P}\left[\,\overline{A_j} \,|\, \overline{A_i}\,\right]\right).
\end{aligned}$$

Clearly, $\mathrm{P}\left[\overline{A_i}\right] = 4^{-k}$. Moreover, for $j < i$, the first $j - i$ letters of S_j^k are independent of the letters of S_i^k, so $\mathrm{P}\left[\overline{A_j} \mid \overline{A_i}\right] \leq \frac{1}{4^{i-j}}$. It follows that $\mathrm{P}[C_i] \geq 4^{-k} \cdot (1 - \frac{1}{3})$, hence

$$\mathrm{P}\left[\bigwedge_{i=1}^{n-k+1} A_i\right] \leq \left(1 - (k+1) \cdot \frac{2}{3} 4^{-k}\right)^{\lfloor n/2k \rfloor} \leq e^{-(k+1)\frac{2}{3}4^{-k} \cdot \lfloor n/2k \rfloor} \leq e^{-\frac{1}{3}n/4^k}. \quad \blacksquare$$

■

Theorem 6. *If $n = \Omega(\log k \cdot 4^k)$ then every algorithm for RBH fails with probability of at least $\frac{2}{3}$, even when the homologous sequence differs from the target sequence in one letter (whose position is known).*

Proof. Suppose that $n \geq 3\ln(36k) \cdot 4^k + 2k - 1$. Let $\{k\}$ be the locations set. For a sequence S and an integer i, let $S[i]$ be the set containing S and the 3 sequences that are obtained from S by changing the i-th letter of S. We say that a sequence S of length n is *hard* if for every sequence $T \in S_1^{2k-1}[k]$, all the k-subsequences of T appear in $S_{2k}^{n-(2k-1)}$. By Lemma 5, the probability that a random sequence S is not hard is at most $4k \cdot e^{-\frac{1}{3}(n-2k+1)/4^k} \leq \frac{1}{9}$. Therefore, it is suffices to bound the success probability on hard sequences.

Let S be a hard sequence. All the sequences in $S[k]$ have the same spectrum, so their corresponding inputs to the RBH problem are equal. Therefore, every algorithm will fail with probability of at least $\frac{3}{4}$ when given a random sequence from $S[k]$. Since this is true for every hard sequence, it follows that every algorithm fails with probability of at least $\frac{3}{4}$ on a random hard sequence, and therefore any algorithm fails with probability of at least $\frac{8}{9} \cdot \frac{3}{4} = \frac{2}{3}$ on a random sequence of length n. ■

5 Hybridization errors

In this section, we study the RBH problem in a more realistic scenario, in which there are errors in the hybridization data. We assume the following model of errors: Each k-tuple contained in the target appears in the (experimental) spectrum with probability $1 - q$, and each k-tuple that is not contained in the target appears in the spectrum with probability p. In other words, the false negative probability is q, and the false positive probability is p. Furthermore, the appearance of a tuple is independent of the other k-tuples.

We say that a sequence S is *simple* if there are no indices $i \neq j$ such that $S_i^k = S_j^k$. Let algorithm B be an algorithm that acts like algorithm A, except that steps 3b and 3c are replaced by the following steps

(3b') Choose a sequences $B \in \mathcal{B}$ such that $S_{t-k+1}^{k-1}B$ is simple, and the number of supporting probes of $S_{t-k+1}^{k-1}B$ is maximal (breaking ties arbitrarily).

Table 1: Performance of algorithm A.

k	$d=1$	$d=2$	$d=3$	$d=4$	$d=5$	Classical SBH
7	1900	1340	980	730	470	120
8	7240	5150	3940	3270	2440	240

Table 2: Performance of algorithm B.

k	$d=1$	$d=2$	$d=3$	$d=4$
7	1240	600	290	100
8	4360	2500	1130	390

(3c') Set set s_t to the first letter of B.

and steps 4b and 4c are replaced by similar steps.

Theorem 7. *For every $\epsilon > 0$ and $c \geq 2$, if $p = O(1/k)$,*

$$d \leq \left(\frac{3}{4} - \sqrt{\frac{1}{c} + \frac{1}{2}\log(1+4q) + O(\log k/k)} \right) k,$$

and $n = O(\epsilon(kd)^{-1} 4^{(1-\frac{c}{4}\log(1+4q))k})$ then the probability that algorithm B fails is at most ϵ.

6 Experimental results

To complement our theoretical results, we performed simulations with our algorithms. For each value of k and d, we run algorithm A on 1000 random sequences of length n for various values of n, and computed the maximum value of n for which algorithm A returned the correct sequence in at least 90% of the runs. The results are given in Table 1. We also performed simulations with algorithm B, using the parameters $p = q = 0.05$. The results are given in Table 2. We note that further research is needed in order to evaluate the performance of algorithm B on real data. Some modification to the algorithm might be needed as real data do not behave like our probabilistic model.

Acknowledgments

We thank Ron Shamir for helpful discussions.

References

[1] L. M. Adleman. Location sensitive sequencing of DNA. Technical report, University of Southern California, 1998.

[2] R. Arratia, D. Martin, G. Reinert, and M. S. Waterman. Poisson process approximation for sequence repeats, and sequencing by hybridization. *J. of Computational Biology*, 3(3):425–463, 1996.

[3] W. Bains and G. C. Smith. A novel method for nucleic acid sequence determination. *J. Theor. Biology*, 135:303–307, 1988.

[4] A. Ben-Dor, I. Pe'er, R. Shamir, and R. Sharan. On the complexity of positional sequencing by hybridization. *J. Theor. Biology*, 8(4):88–100, 2001.

[5] J. Błażewicz, P. Formanowicz, F. Glover, M. Kasprzak, and J. Węglarz. An improved tabu search algorithm for DNA sequencing with errors. In *Proc. 3rd Metaheuristics International Conference*, pages 69–75, 1999.

[6] J. Błażewicz, P. Formanowicz, F. Guinand, and M. Kasprzak. A heuristic managing errors for DNA sequencing. *Bioinformatics*, 18(5):652–660, 2002.

[7] J. Błażewicz, P. Formanowicz, M. Kasprzak, W. T. Markiewicz, and J. Węglarz. DNA sequencing with positive and negative errors. *J. of Computational Biology*, 6(1):113–123, 1999.

[8] J. Błażewicz, P. Formanowicz, M. Kasprzak, W. T. Markiewicz, and J. Węglarz. Tabu search for dna sequencing with false negatives and false positives. *European Journal of Operational Research*, 125:257–265, 2000.

[9] J. Błażewicz, J. Kaczmarek, M. Kasprzak, W. T. Markiewicz, and J. Węglarz. Sequential and parallel algorithms for DNA sequencing. *CABIOS*, 13:151–158, 1997.

[10] J. Błażewicz, M. Kasprzak, and W. Kuroczycki. Hybrid genetic algorithm for DNA sequencing with errors. *J. of Heuristics*, 8:495–502, 2002.

[11] S. D. Broude, T. Sano, C. S. Smith, and C. R. Cantor. Enhanced DNA sequencing by hybridization. *Proc. Nat. Acad. Sci. USA*, 91:3072–3076, 1994.

[12] R. Drmanac, I. Labat, I. Brukner, and R. Crkvenjakov. Sequencing of megabase plus DNA by hybridization: theory of the method. *Genomics*, 4:114–128, 1989.

[13] M. E. Dyer, A. M. Frieze, and S. Suen. The probability of unique solutions of sequencing by hybridization. *J. of Computational Biology*, 1:105–110, 1994.

[14] A. Frieze, F. Preparata, and E. Upfal. Optimal reconstruction of a sequence from its probes. *J. of Computational Biology*, 6:361–368, 1999.

[15] A. M. Frieze and B. V. Halldórsson. Optimal sequencing by hybridization in rounds. *J. of Computational Biology*, 9(2):355–369, 2002.

[16] E. Halperin, S. Halperin, T. Hartman, and R. Shamir. Handling long targets and errors in sequencing by hybridization. In *Proc. 6th Annual International Conference on Computational Molecular Biology (RECOMB '02)*, pages 176–185, 2002.

[17] S. Hannenhalli, P. A. Pevzner, H. Lewis, and S. Skiena. Positional sequencing by hybridization. *Computer Applications in the Biosciences*, 12:19–24, 1996.

[18] S. A. Heath and F. P. Preparata. Enhanced sequence reconstruction with DNA microarray application. In *COCOON '01*, pages 64–74, 2001.

[19] S. A. Heath, F. P. Preparata, and J. Young. Sequencing by hybridization using direct and reverse cooperating spectra. In *Proc. 6th Annual International Conference on Computational Molecular Biology (RECOMB '02)*, pages 186–193, 2002.

[20] H. W. Leong, F. P. Preparata, W. K. Sung, and H. Willy. On the control of hybridization noise in DNA sequencing-by-hybridization. In *Proc. 2nd Workshop on Algorithms in Bioinformatics (WABI '02)*, pages 392–403, 2002.

[21] R. J. Lipshutz. Likelihood DNA sequencing by hybridization. *J. Biomolecular Structure and Dynamics*, 11:637–653, 1993.

[22] Y. Lysov, V. Floretiev, A. Khorlyn, K. Khrapko, V. Shick, and A. Mirzabekov. DNA sequencing by hybridization with oligonucleotides. *Dokl. Acad. Sci. USSR*, 303:1508–1511, 1988.

[23] D. Margaritis and S. Skiena. Reconstructing strings from substrings in rounds. In *Proc. 36th Symposium on Foundation of Computer Science (FOCS 95)*, pages 613–620, 1995.

[24] I. Pe'er, N. Arbili, and R. Shamir. A computational method for resequencing long dna targets by universal oligonucleotide arrays. *Proc. National Academy of Science USA*, 99:15497–15500, 2002.

[25] I. Pe'er and R. Shamir. Spectrum alignment: Efficient resequencing by hybridization. In *Proc. 8th International Conference on Intelligent Systems in Molecular Biology (ISMB '00)*, pages 260–268, 2000.

[26] P. A. Pevzner. l-tuple DNA sequencing: Computer analysis. *J. Biomolecular Structure and Dynamics*, 7:63–73, 1989.

[27] P. A. Pevzner, Yu. P. Lysov, K. R. Khrapko, A. V. Belyavsky, V. L. Florentiev, and A. D. Mirzabekov. Improved chips for sequencing by hybridization. *J. Biomolecular Structure and Dynamics*, 9:399–410, 1991.

[28] F. Preparata and E. Upfal. Sequencing by hybridization at the information theory bound: an optimal algorithm. In *Proc. 4th Annual International Conference on Computational Molecular Biology (RECOMB '00)*, pages 88–100, 2000.

[29] R. Shamir and D. Tsur. Large scale sequencing by hybridization. *J. of Computational Biology*, 9(2):413–428, 2002.

[30] S. Skiena and S. Snir. Restricting SBH ambiguity via restriction enzymes. In *Proc. 2nd Workshop on Algorithms in Bioinformatics (WABI '02)*, pages 404–417, 2002.

[31] S. Skiena and G. Sundaram. Reconstructing strings from substrings. *J. of Computational Biology*, 2:333–353, 1995.

[32] S. Snir, E. Yeger-Lotem, B. Chor, and Z. Yakhini. Using restriction enzymes to improve sequencing by hybridization. Technical Report CS-2002-14, Technion, Haifa, Israel, 2002.

[33] D. Tsur. Sequencing by hybridization with errors: Handling longer sequences. Manuscript, 2003.

[34] D. Tsur. Sequencing by hybridization in few rounds. In *Proc. ESA '03*, to appear.