Sequencing by Hybridization with Errors: Handling Longer Sequences

Dekel Tsur[*]

Abstract

Sequencing by Hybridization (SBH) is a method for reconstructing a DNA sequence given the set of all subsequences of length k of the target sequence. This set, called the *spectrum* of the sequence, can be obtained from hybridization with a universal DNA chip. However, the hybridization experiments are error prone, so this leads to the computational problem of reconstructing a sequence from a noisy spectrum. Halperin et al. gave an algorithm for this problem with provable performance in the presence of both false positive and false negative errors. Assuming, for example, that the false positive rate is small, and the probability of false negative is 0.1, the algorithm can reconstruct a random sequence of length $O(2^{0.7k})$ with an arbitrary small probability of failure. In this paper, we give an algorithm that can reconstruct longer sequences: Under the assumptions above, our algorithm can reconstruct sequences of length $O(2^{0.942k})$. This bound is almost optimal as the bound for the errorless case is $\Theta(2^k)$.

1 Introduction

Sequencing by Hybridization (SBH) [3, 14] is a method for sequencing DNA fragments. In this method, the target sequence is hybridized to a universal chip containing all 4^k sequences of length k. Each sequence in the chip whose reverse complement appears in the target will hybridize to the target, and this hybridization can be detected. Thus, one can obtain the set of all subsequences of length k of a target sequence. This set is called the *k-spectrum* (or *spectrum*) of the target.

Clearly, different sequences can have the same spectrum. It is known that if the target sequence is chosen uniformly from the set of all sequences of length n for $n = O(2^k)$, then with probability close to 1, there is no other sequence of length n with the same spectrum as the target's [16]. Thus, sequences of length $O(2^k)$ can be reconstructed with small probability of failure, and this bound is asymptotically optimal [1, 2, 10, 17].

In practice, the hybridization experiments are error prone. In a *false positive* error, a certain k-tuple appears in the experimental spectrum while in fact it does not appear in the

[*] Caesarea Rothschild Institute of Computer Science, University of Haifa. Email: dekelts@cs.haifa.ac.il

target. The converse occurs in a *false negative* error. The problem of reconstructing the sequence when there are hybridization errors is NP-hard [11]. However, several heuristics were proposed [4–9,13,15]. Halperin et al. [12] gave an algorithm with provable performance in the following model: 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. Halperin et al. proved that if $p < \frac{1}{2^k}$, then the algorithm can reconstruct a random sequence of length $O(2^{(1-3q)k})$ from its k-spectrum, with an arbitrary small probability of failure.

In this paper, we give an algorithm that can reconstruct longer sequences than the algorithm of Halperin et al.: Under the same model as above, the algorithm can reconstruct sequences of length $O(2^{(1-\beta-\delta)k})$, where $\beta = \alpha/(\alpha + \log_2(1/q))$, $\alpha = \log_4(1 + 3q + q/4 \cdot (1 - q)/(1 - q/4))$, and δ is an arbitrary small constant. Moreover, our algorithm requires only that p is smaller than some constant that depends on q and δ. Note that $\beta < 3q$ for every $q > 0$ (for example, for $q = 0.1$, $\beta \approx 0.057 < 0.3$), so our algorithm performs better than the algorithm of Halperin et al. for every q.

We finish this section with some definitions. For a sequence $S = s_1 \cdots s_n$, S_i^l is the subsequence $s_i s_{i+1} \cdots s_{i+l-1}$ of S. Fix some k. We say that a sequence S is *simple* if there are no indices $i \neq j$ such that $|i - j| < k$ and $S_i^k = S_j^k$. A sequence S is *strongly simple* if there are no two indices $i \neq j$ such that $|i - j| \leq 4k$ and $S_i^{\lceil \frac{2}{3}k \rceil} = S_j^{\lceil \frac{2}{3}k \rceil}$. For simplicity, we assume in the following that k is divisible by 3.

2 The algorithm

In the rest of the paper, we shall use $A = a_1 \cdots a_n$ to denote the target sequence. Given the (experimental) spectrum of the target sequence, a *supporting probe* for a sequence S is a sequence of length k that appears in S and in the spectrum.

Let $l = \lceil k/(\alpha + \log_2(1/q)) \rceil$. Note that $l \leq k$ for every $q > 0$. We assume that the first and last $k - 1$ letters of A are known. The reconstruction algorithm is as follows:

1. Set $i = k$.
2. Enumerate all simple sequences of length l.
3. Pick a simple sequence $B' = b_1, \ldots, b_l$ such that the number of supporting probes for $s_{i-k+1} \cdots s_{i-1} b_1 \cdots b_l$ is maximal (breaking ties arbitrarily).
4. Set $s_i = b_1$.
5. If $i < n - k + 1$, increase i by 1 and goto 2.

A sequence of length l that is constructed in step 2 of the algorithm is called a *path (w.r.t. i)*. The path $a_i \cdots a_{i+l-1}$ will be called the *correct path (w.r.t. i)*. A path is called *bad* if its first letter is not equal to a_i.

Note that our algorithm is similar to the algorithm of Halperin et al. [12]. The main difference is that our algorithm uses paths of length $l \leq k$, while the algorithm of Halperin et al. uses paths of length k. The motivation behind this difference is that when the paths have length k, it is more likely that one of the probes that should support the correct path will not appear in the spectrum, so the probability of failure increases. Another difference is that our algorithm only considers simple paths. This fact simplifies the analysis of the algorithm.

Theorem 1. *For every $0 < \delta < 1$, if $p \leq \min(\frac{1}{2}q, 16^{-5\log_2(1/\delta)/\delta}, (1-q)\delta/8)$ and $n = O(2^{(1-\beta-\delta)k})$, then the probability that the algorithm fails is $o(1)$.*

Proof. Fix some δ. Suppose that the algorithm fails, and let t be the minimum index such that $s_t \neq a_t$. Let X be a random variable that counts the number of supporting probes for the correct path (w.r.t. t). Define the following events:

(E_0) The target sequence is not simple.

(E_1) The target sequence is not strongly simple.

(E_2) $X \leq \delta' l$, where $\delta' = \frac{1}{5}\delta / \log_2(e/\delta)$.

(E_3) There is a bad path (w.r.t. t) with at least X supporting probes.

Since the algorithm failed to reconstruct a_t, we must have that either the correct path lost to some bad path in step 3, namely event E_3 occurs, or the correct path was not considered by the algorithm as it is not simple. In the latter case, we have that event E_0 occurs. Therefore, the probability that the algorithm fails is at most $\mathrm{P}[E_0 \vee E_3]$. We have that

$$\mathrm{P}[E_0 \vee E_3] \leq \mathrm{P}[E_1 \vee E_3] \leq \mathrm{P}[E_1] + \mathrm{P}\left[E_2 | \overline{E_1}\right] + \mathrm{P}\left[E_3 | \overline{E_1} \wedge \overline{E_2}\right].$$

We shall show that each of the last three probabilities is $o(1)$. The reason why we consider the events E_1 and E_2 is that it is easier to estimate $\mathrm{P}\left[E_3 | \overline{E_1} \wedge \overline{E_2}\right]$ than to estimate $\mathrm{P}[E_3]$ directly.

Given two indices $i < j$, the probability that $A_i^{\frac{2}{3}k} = A_j^{\frac{2}{3}k}$ is exactly $4^{-\frac{2}{3}k}$ (this is true even when $|i-j| < k$). The number of ways to choose the indices i and j is at most $4kn$. Therefore, $\mathrm{P}[E_1] \leq 4kn4^{-\frac{2}{3}k} = o(1)$.

We now consider event E_2. As we assume that event E_1 does not happen, we have that X has binomial distribution with l experiments and success probability $1-q$, so

$$\mathrm{P}[X \leq \delta' l] = \mathrm{P}[l - X \geq (1-\delta')l] \leq \binom{l}{\delta' l} q^{(1-\delta')l}.$$

Claim 2. $q^{(1-\delta')l} \leq 2^{-(1-\beta-\frac{1}{5}\delta)k}$.

Proof. From the definitions of l and δ' we have

$$\begin{aligned} q^{(1-\delta')l} &= 2^{-\log_2(1/q)\cdot(1-\delta')l} \le 2^{\log_2(1/q)\cdot(1-\delta')\cdot k/(\alpha+\log_2(1/q))} = 2^{-(1-\beta)(1-\delta')k} \\ &\le 2^{-(1-\beta-\delta')k} \le 2^{-(1-\beta-\frac{1}{5}\delta)k}. \end{aligned}$$ ■

Using Claim 2 and the inequality $\binom{a}{b} \le \left(\frac{ea}{b}\right)^b$ we obtain that

$$\mathrm{P}\left[X \le \delta' l\right] \le \left(\frac{e}{\delta'}\right)^{\delta' l} q^{(1-\delta')l} = 2^{\delta' \log_2(e/\delta')\cdot l} q^{(1-\delta')l} \le 2^{-(1-\beta-\frac{4}{5}\delta)k}.$$

To bound $\mathrm{P}\left[E_2|\overline{E_1}\right]$, we multiply the probability above by the number of ways to choose t, which is at most n. Thus, $\mathrm{P}\left[E_2|\overline{E_1}\right] \le n2^{-(1-\beta-4/5\cdot\delta)k} = o(1)$.

We now bound the probability of event E_3. We select a bad path $b_1\cdots b_l$ at random, and let Y be the number of probes supporting this path. Let P_{bad} be the probability that $Y \ge X$ assuming that $X \ge \delta' l$. Clearly, $Y = Y_1 + Y_2$, where Y_1 is the number of supporting probes for the bad path that appear in the target, and Y_2 is the number of supporting probes arising from false positives. Let Y_0 denote the number of sequences of length k that appear both in $s_{t-k+1}\cdots s_{t-1}b_1\cdots b_l$ and in the target (but not necessarily in the spectrum).

We will bound the probability that $Y = i$. Clearly, $\mathrm{P}\left[Y = i\right] = \sum_{a=0}^{i} f(a)$, where

$$f(a) = \mathrm{P}\left[Y = i|Y_2 = a\right] = \sum_{j=i-a}^{l} \mathrm{P}\left[Y_0 = j\right] \mathrm{P}\left[Y_1 = i-a|Y_0 = j\right] \mathrm{P}\left[Y_2 = a|Y_0 = j\right].$$

Moreover, $\mathrm{P}\left[Y_1 = i-a|Y_0 = j\right] = \binom{j}{i-a} q^{j-(i-a)}(1-q)^{i-a}$ and

$$\mathrm{P}\left[Y_2 = a|Y_0 = j\right] = \binom{l-j}{a} p^a (1-p)^{l-j-a} \le \binom{l}{a} p^a.$$

A bound on the probability that $Y_0 = j$ is given by the following lemma, which is similar to Lemma 3.2 in [12]. We note that some details are missing in the proof in [12], while we give here a complete proof.

Lemma 3. *For $j > 0$,* $\mathrm{P}\left[Y_0 = j\right] \le 5nl \cdot 4^{-(k+j)}$.

Proof. Denote $B = s_{t-k+1}\cdots s_{t-1}b_1\cdots b_l$. If $Y_0 = j$ then there is a set $I \subseteq \{1,\ldots,l\}$ of size j and indices $\{r_i : i \in I\}$ such that $B_i^k = A_{r_i}^k$ for $i \in I$. The sequence $A_{r_i}^k$ will be called *probe* i. Note that $r_i \ne t-1+i$ for all $i \in I$ as $b_1 \ne a_t$. We say that probes i and i' $(i, i' \in I)$ are *adjacent* if $r_i - r_{i'} = i - i'$ (in particular, every probe is adjacent to itself). For two adjacent probes i and i', with $i < i'$, we have that $B_i^k = A_{r_i}^k$ and $B_{i'}^k = A_{r_{i'}}^k$ if and only if $B_i^{k+i'-i} = A_{r_i}^{k+i'-i}$.

We can assume w.l.o.g. that each equivalence class of the adjacency relation is an interval in I, and let $I_1,\ldots,I_x \subseteq I$ be the equivalence classes, where $\min(I_1) < \min(I_2) < \cdots < \min(I_x)$. We have that $B_i^k = A_{r_i}^k$ for all $i \in I$ if and only if $B_{\min(I_i)}^{k-1+|I_i|} = A_{r_{\min(I_i)}}^{k-1+|I_i|}$ for $i = 1,\ldots,x$. Each sequence $A_{r_{\min(I_i)}}^{k-1+|I_i|}$ will be called a *block*, and will be denoted by L_i. We also define L_0 to be the sequence A^k_{t-k+1}. A block L_i is called *overlapping* if there is an index $i' < i$ such that $|r_{\min(I_i)} - r_{\min(I_{i'})}| \leq 4k-4$, and let y the the index of the first overlapping block, if there is such a block. Note that block L_y shares letters with at most one block L_i with $i < y$. We consider 3 cases, which will be denoted $\mathcal{E}_1$, $\mathcal{E}_2$, and $\mathcal{E}_3$:

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

Case 1 For fixed I and $\{r_i : i \in I\}$, the probability that $\mathcal{E}_1$ happens is $\prod_{i=1}^x 4^{-(k-1+|I_i|)} = 4^{-(k-1)x-j}$. The number of ways to choose disjoint (non empty) intervals $I_1, \ldots, I_x \subseteq \{1, \ldots, l\}$ such that $\sum_{i=1}^x |I_i| = j$ is $\binom{j-1}{x-1}\binom{l-j+x}{x} \leq \binom{j-1}{x-1} l^x$. For a fixed choice of $I_1, \ldots, I_x$, there are at most n^x ways to choose the indices $\{r_i : i \in I\}$. Therefore,

$$\begin{aligned}
\mathrm{P}\left[\mathcal{E}_1\right] &\leq \sum_{x=1}^{j} \binom{j-1}{x-1} (nl)^x \frac{1}{4^{(k-1)x+j}} = \frac{nl}{4^{k-1+j}} \sum_{x=1}^{j} \binom{j-1}{x-1} \left(\frac{nl}{4^{k-1}}\right)^{x-1} \\
&= \frac{nl}{4^{k-1+j}} \left(1 + \frac{nl}{4^{k-1}}\right)^{j-1} \leq \frac{nl}{4^{k-1+j}} e^{jnl/4^{k-1}} = (1+o(1)) \frac{nl}{4^{k-1+j}}.
\end{aligned}$$

Case 2 Let $\mathcal{E}$ be the event that $B^{k-1+|I_i|}_{\min(I_i)} = A^{k-1+|I_i|}_{r_{\min(I_i)}}$ for $i = 1, \ldots, y-1$, and let $\mathcal{E}'$ be the event that $B^k_{\min(I_y)} = A^k_{r_{\min(I_y)}}$. Let $z = 1 + \sum_{i=1}^{y-1} |I_i|$ and $I' = I \cap \{1, \ldots, \min(I_y)\}$. For fixed I' and $\{r_i : i \in I'\}$, the probability that event $\mathcal{E}$ happens is $4^{-(k-1)(y-1)-(z-1)}$ and the probability that event $\mathcal{E}'$ happens is 4^{-k}. Moreover, events $\mathcal{E}$ and $\mathcal{E}'$ are independent (see [18]). The number of ways to choose the intervals $I_1, \ldots, I_{y-1}$ and $\min(I_y)$ is $\binom{z-2}{y-2}\binom{l-z+y}{y} \leq \binom{(z-1)-1}{(y-1)-1} l^y$. For fixed $I_1, \ldots, I_{y-1}$ and $\min(I_y)$, there are at most $n^{y-1} \cdot 8k(y-1) \leq n^{y-1} \cdot 8kl$ ways to choose the indices $\{r_i : i \in I'\}$ (as $|r_{\min(I_y)} - r_{\min(I_i)}| \leq 4k-4$ for some $i < y$). Thus,

$$\begin{aligned}
\mathrm{P}\left[\mathcal{E} \wedge \mathcal{E}' | z\right] &\leq 8kl^2 \sum_{y=2}^{z} \binom{z-2}{y-2} (nl)^{y-1} \frac{1}{4^{(k-1)(y-1)+z-1+k}} \\
&= \frac{8nkl^3}{4^{2k+z-2}} \sum_{y=2}^{z} \binom{z-2}{y-2} \left(\frac{nl}{4^{k-1}}\right)^{y-2} \\
&\leq \frac{8nkl^3}{4^{2k+z-2}} e^{znl/4^{k-1}} = (1+o(1)) \frac{8nkl^3}{4^{2k+z-2}}.
\end{aligned}$$

If $j < \frac{2}{3}k$ then

$$\mathrm{P}\left[\mathcal{E}_2\right] \leq \mathrm{P}\left[\mathcal{E} \wedge \mathcal{E}'\right] = O\left(\frac{nkl^3}{4^{2k}} \sum_{z=2}^{j} \frac{1}{4^z}\right) = O\left(\frac{nk^4}{4^{2k}}\right) = o\left(\frac{n}{4^{k+j}}\right).$$

Now, consider the case when $j \geq \frac{2}{3}k$. If $y = x$ then the analysis is the same as the analysis of case 1, as event $\mathcal{E}$ and the event that $B^{k-1+|I_x|}_{\min(I_x)} = A^{k-1+|I_x|}_{r_{\min(I_x)}}$ are independent. We therefore assume that $y < x$.

Let $\mathcal{E}''$ be the event that $B^k_{\max(I_x)} = A^k_{r_{\max(I_x)}}$. If $z \leq \frac{1}{3}k$ then we have that $\max(I_x) - \min(I_y) \geq j - z \geq \frac{1}{3}k$. It follows that the last $\frac{1}{3}k$ letters of $B^k_{\max(I_x)}$ are not letters of $B^{k-1+|I_1|}_{\min(I_1)}, \ldots, B^{k-1+|I_{y-1}|}_{\min(I_{y-1})}$ or $B^k_{\min(I_y)}$, and thus these letters are not restricted by events $\mathcal{E}$ and $\mathcal{E}'$. Therefore, $\mathrm{P}\left[\mathcal{E}''|\mathcal{E} \wedge \mathcal{E}', z \leq \frac{1}{3}k\right] \leq 4^{-\frac{1}{3}k}$. We conclude that

$$\begin{aligned}\mathrm{P}[\mathcal{E}_2] \leq \mathrm{P}[\mathcal{E} \wedge \mathcal{E}' \wedge \mathcal{E}''] &= O\left(\frac{nkl^3}{4^{2k}}\left(\sum_{z=2}^{\frac{1}{3}k} \frac{1}{4^{z+\frac{1}{3}k}} + \sum_{z=\frac{1}{3}k+1}^{j} \frac{1}{4^z}\right)\right) \\ &= O\left(\frac{nk^4}{4^{\frac{7}{3}k}}\right) = o\left(\frac{n}{4^{k+j}}\right).\end{aligned}$$

Case 3 As L_1 overlaps with L_0, we have that $\min(I_1) > \frac{1}{3}k$ because otherwise we get a contradiction to the assumption that A is strongly simple. Thus, $j \leq \frac{2}{3}k$. Assume again that $y < x$. We consider the events $\mathcal{E}'$ and $\mathcal{E}''$ defined above. If L_x does not overlap with L_1, then these events are independent, so

$$\mathrm{P}[\mathcal{E}_3 | L_x \text{ does not overlap } L_1] \leq \mathrm{P}[\mathcal{E}' \wedge \mathcal{E}''] \leq \frac{8nkl^2}{4^{2k}} = o\left(\frac{n}{4^{k+j}}\right).$$

Otherwise, since $B^k_{\max(I_x)}$ contains at least $j-1$ letters that are not letters of $B^k_{\min(I_1)}$, it follows that

$$\mathrm{P}[\mathcal{E}_3 | L_x \text{ overlaps } L_1] = O\left(\frac{k^2 l^2}{4^{k+j}}\right) = o\left(\frac{n}{4^{k+j}}\right).$$

Combining the three cases, we have that $\mathrm{P}[Y_0 = j] = (1 + o(1))4nl \cdot 4^{-(k+j)}$. ■

By differentiating the identity $\sum_{b=0}^{\infty} x^b = \frac{1}{1-x}$ (for $x < 1$) y times we get that $\sum_{b=0}^{\infty} \binom{y+b}{y} x^b = \frac{1}{(1-x)^{y+1}}$. Using the latter identity and Lemma 3, we obtain that for $a < i$,

$$\begin{aligned} f(a) &\leq \sum_{j=i-a}^{l} \frac{5nl}{4^{k+j}} \binom{j}{i-a} q^{j-(i-a)} (1-q)^{i-a} \binom{l}{a} p^a \\ &= \frac{5nl}{4^{k+i-a}} (1-q)^{i-a} \binom{l}{a} p^a \sum_{b=0}^{l-(i-a)} \binom{i-a+b}{i-a} \left(\frac{q}{4}\right)^b \\ &\leq \frac{5nl}{4^{k+i-a}} (1-q)^{i-a} \binom{l}{a} p^a \frac{1}{(1-\frac{q}{4})^{i-a+1}} \\ &= \frac{5nl}{(1-\frac{q}{4})4^k} \left(\frac{1-q}{4(1-\frac{q}{4})}\right)^i \binom{l}{a} \left(p \cdot \frac{4(1-\frac{q}{4})}{1-q}\right)^a. \end{aligned}$$

Furthermore,

$$f(i) \le \sum_{j=0}^{l} \mathrm{P}\left[Y_0 = j\right] \binom{l}{i} p^i \le 2^l p^i \sum_{j=0}^{l} \mathrm{P}\left[Y_0 = j\right] \le 2^l p^i.$$

Therefore,

$$\begin{aligned}
\mathrm{P}\left[Y = i\right] &\le 2^l p^i + \frac{5nl}{(1-\frac{q}{4})4^k} \left(\frac{1-q}{4(1-\frac{q}{4})}\right)^i \cdot \sum_{a=0}^{i-1} \binom{l}{a} \left(p \cdot \frac{4(1-\frac{q}{4})}{1-q}\right)^a \\
&\le 2^l p^i + \frac{7nl}{4^k} \left(\frac{1-q}{4(1-\frac{q}{4})}\right)^i \left(1 + p \cdot \frac{4(1-\frac{q}{4})}{1-q}\right)^l \\
&\le 2^l p^i + \frac{7nl}{4^k} \left(\frac{1-q}{4(1-\frac{q}{4})}\right)^i e^{p \cdot \frac{4(1-\frac{q}{4})}{1-q} \cdot l} \\
&\le 2^l p^i + \frac{7nl}{4^k} \left(\frac{1-q}{4(1-\frac{q}{4})}\right)^i 4^{\frac{1}{2}\delta l}.
\end{aligned}$$

Now, $\mathrm{P}\left[X \le i\right] \le \binom{l}{i} q^{l-i}$. Hence,

$$\begin{aligned}
P_{\text{bad}} &\le \sum_{i=\delta' l}^{l} \mathrm{P}\left[X \le i\right] \cdot \mathrm{P}\left[Y = i\right] \\
&\le \sum_{i=\delta' l}^{l} \binom{l}{i} q^{l-i} \cdot \frac{7nl}{4^{k-\frac{1}{2}\delta l}} \left(\frac{1-q}{4(1-\frac{q}{4})}\right)^i + \sum_{i=\delta' l}^{l} \binom{l}{i} q^{l-i} \cdot 2^l p^i.
\end{aligned}$$

We denote the two sums above by S_1 and S_2. Then,

$$\begin{aligned}
S_1 &\le \frac{7nl}{4^{k-\frac{1}{2}\delta l}} \sum_{i=0}^{l} \binom{l}{i} q^{l-i} \left(\frac{1-q}{4(1-\frac{q}{4})}\right)^i = \frac{14nl}{4^{k-\frac{1}{2}\delta l}} \left(q + \frac{1-q}{4(1-\frac{q}{4})}\right)^l \\
&= \frac{7nl}{4^{k-\frac{1}{2}\delta l}} \left(\frac{1 + 3q + \frac{q(1-q)}{4(1-q/4)}}{4}\right)^l = \frac{14nl}{4^l} \cdot \frac{1}{4^{k-\alpha l-\frac{1}{2}\delta l}} \\
&\le \frac{7nl}{4^l} \cdot \frac{1}{4^{k-\alpha(k/(\log_2(1/q)+\alpha)-1)-\frac{1}{2}\delta k}} = \frac{14nl}{4^l} \cdot \frac{4^\alpha}{4^{k-\beta k-\frac{1}{2}\delta k}} \\
&\le \frac{7nl}{4^l} \cdot \frac{4}{4^{(1-\beta-\frac{1}{2}\delta)k}},
\end{aligned}$$

and by Claim 2,

$$\begin{aligned}
S_2 &\le \sum_{i=\delta' l}^{l} \binom{l}{i} q^{l-i} \cdot 2^l p^i \le 4^l \sum_{i=\delta' l}^{l} q^{l-i} p^i = 4^l p^{\delta' l} q^{(1-\delta')l} \sum_{a=0}^{(1-\delta')l} \left(\frac{p}{q}\right)^a \\
&\le 4^l p^{\delta' l} 2^{-(1-\beta-\frac{1}{5}\delta)k} \sum_{a=0}^{(1-\delta')l} \frac{1}{2^a} \le 4^l p^{\delta' l} 2^{-(1-\beta-\frac{1}{5}\delta)k} \cdot 2.
\end{aligned}$$

The probability that event E_3 happens (given that E_1 and E_2 do not happen) is at most $n4^l P_{\text{bad}}$, where n bounds the number of ways to choose t, and 4^l bounds the number of ways to choose a bad path. We have that

$$n4^l S_1 \leq 28l \cdot \left(\frac{n}{2^{(1-\beta-\frac{1}{2}\delta)k}} \right)^2 = o(1)$$

and

$$n4^l S_2 \leq 2n \cdot \left(16^{1/\delta'} p \right)^{\delta' l} \cdot 2^{-(1-\beta-\frac{1}{5}\delta)k} \leq 2n \cdot 2^{-(1-\beta-\frac{1}{5}\delta)k} = o(1).$$

Therefore, $\mathrm{P}\left[E_3 | \overline{E_1} \wedge \overline{E_2}\right] = o(1)$. ■

References

[1] R. Arratia, B. Bollobás, D. Coppersmith, and G. Sorkin. Euler circuits and DNA sequencing by hybridization. *Discrete Applied Math*, 104:63–96, 2000.

[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] 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.

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

[6] 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.

[7] 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.

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

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

[10] 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.

[11] J. Gallant, D. Maier, and J. A. Storer. On finding minimal length superstrings. *J. of Computer and System Sciences*, 20:50–58, 1980.

[12] 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.

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

[14] 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.

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

[16] P. A. Pevzner and R. J. Lipshutz. Towards DNA sequencing chips. In *Symp. on Mathematical Foundations of Computer Science*, LNCS 841, pages 143–158, 1994.

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

[18] D. Tsur. Bounds for resequencing by hybridization. In *Proc. 3rd Workshop on Algorithms in Bioinformatics (WABI '03)*, LNCS 2812, pages 498–511, 2003.