Sequencing by Hybridization in Few Rounds

Dekel Tsur*

Abstract

Sequencing by Hybridization (SBH) is a method for reconstructing an unknown DNA string based on obtaining, through hybridization experiments, whether certain short strings appear in the target string. Following Margaritis and Skiena [12], we study the SBH in rounds problem: The goal is to reconstruct an unknown string A (over a fixed alphabet) using queries of the form "does the string S appear in A?" for some query string S. The queries are performed in rounds, where the queries in each round depend on the answers to the queries in the previous rounds. We show that almost all strings of length n can be reconstructed in $\log^* n$ rounds with $O(n)$ queries per round.

We also consider a variant of the problem in which for each substring query S, the answer is whether S appears once in the string A, appears at least twice in A, or does not appear in A. For this problem, we show that almost all strings can be reconstructed in 2 rounds of $O(n)$ queries. Our results improve the previous results of Margaritis and Skiena [12] and Frieze and Halldórsson [8]. Moreover, the second result is optimal.

Keywords: Sequencing by hybridization; Probabilistic analysis

1 Introduction

Sequencing by Hybridization (SBH) [2, 11] is a method for sequencing of long DNA molecules. In this method, the target string is hybridized to a chip containing known strings. For each string in the chip, if its reverse complement appears in the target, then the two strings will bind (or hybridize), and this hybridization can be detected. Thus, SBH can be modeled as the problem of finding an unknown target string using queries of the form "does S appear in the target string?" for some query string S. Classical SBH consists of making queries for all the strings of length k for some fixed k, and then constructing the target string using the answers to the queries.

Unfortunately, string reconstruction is often not unique: Other strings can have the same set of k-long substrings as the target's. For an alphabet of size σ, only strings of length $O(\sigma^{\frac{1}{2}k})$ can be reconstructed reliably when using queries of length k [1, 6, 15, 17].

*Department of Computer Science, Ben-Gurion University of the Negev.
E-Mail: dekelts@cs.bgu.ac.il

In other words, in order to reconstruct a string of length n, it is required to take $k = 2\log_\sigma n + \Theta(1)$, and thus the number of queries is $\Theta(n^2)$. As this number is large even for short strings, SBH is not considered competitive in comparison with standard gel-based sequencing technologies.

Several methods for overcoming the limitations of SBH were proposed: alternative chip designs [7, 10, 15, 21], using analog spectra [16], using location information [3–5, 9, 17], using a known homologous string [14], and using restriction enzymes [18].

Margaritis and Skiena [12] suggested asking the queries in several rounds, where the queries in each round depend on the answers to the queries in the previous rounds. The goal is to reconstruct the target strings using as few rounds as possible, where each round contains as few queries as possible. Margaritis and Skiena [12] gave several results, including an algorithm that reconstructs *almost* all strings of length n in $O(\log n)$ rounds, where the number of queries in each round is $O(n)$. They also showed that every string of length n can be reconstructed in $O(\log n)$ rounds using $n^2/\log n$ queries in each round. Skiena and Sundaram [19] showed that every string can be reconstructed in $(\sigma - 1)n + O(\sqrt{n})$ rounds with one query per round.

Frieze and Halldórsson [8] studied a variant of the problem, in which for each substring query, the answer is whether the string appears once in the target, appears at least twice in the target, or does not appear in the target. We call this model the *ternary model*, while the former model will be called the *binary model*. For the ternary model, Frieze and Halldórsson gave an algorithm that reconstruct almost all strings of length n in 7 rounds with $O(n)$ queries in each round.

There are several known lower bounds on the string reconstruction problem. First, in order to reconstruct a constant fraction of all strings of length n, a total of $\Omega(n)$ queries are needed [7]. Moreover, if only one round of queries is performed, then $\Omega(n^2)$ queries are needed [20]. These upper bound apply both to the binary and ternary models.

In this paper, we investigate the string reconstruction problem when the number of queries in each round is linear in the length of the target string. We improve the results of [12] and [8] as follows: For the binary model, we show that almost all strings of length n can be reconstructed in $\log^*_\sigma n$ rounds (using $O(n)$ queries per round). For the ternary model, we show that almost all strings of length n can be reconstructed in 2 rounds (using $O(n)$ queries per round). The latter result is optimal due to the lower bounds mentioned above.

We note that for obtaining our results, we use the algorithms from [8, 12] with some changes in the parameters of the algorithms. The contribution of this paper is a new analysis which gives the reduction in the number of rounds.

The rest of this paper is organized as follows: Section 2 contains basic definitions and top-level description of our algorithms. In Section 3 we give the algorithm for the binary model, and in Section 4 we give the algorithm for the ternary model.

2 Preliminaries

For clarity, we shall assume for the rest of the paper that the alphabet of the strings is $\Sigma = \{A, C, G, T\}$. However, our results hold for any finite alphabet.

For a string $A = a_1 \cdots a_n$, let A_i^l denote the l-substring $a_i a_{i+1} \cdots a_{i+l-1}$. The *binary k-spectrum* of a string A is a mapping $\mathrm{SP}_2^{A,k} \colon \Sigma^k \to \{0, 1\}$ such that $\mathrm{SP}_2^{A,k}(B) = 1$ if B is a substring of A, and $\mathrm{SP}_2^{A,k}(B) = 0$ otherwise. The *ternary k-spectrum* of A is a mapping $\mathrm{SP}_3^{A,k} \colon \Sigma^k \to \{0, 1, 2\}$, where $\mathrm{SP}_3^{A,k}(B) = 0$ if B is not a substring of A, $\mathrm{SP}_3^{A,k}(B) = 1$ if B appears in A exactly once, and $\mathrm{SP}_3^{A,k}(B) = 2$ if B appears in A twice or more. We shall omit the subscript when referring to a spectrum of unspecified type, or when the type of the spectrum is clear from the context.

Let $\log n = \log_4 n$, $\log^{(1)} n = \log n$ and $\log^{(i)} n = \log(\log^{(i-1)} n)$ for $i > 1$. Define $\log^* n$ to be the minimum integer i such that $\log^{(i)} n \leq 1$.

As mentioned in the introduction, we are interested in algorithms that reconstruct almost all strings of some length n. It is convenient to assume that the target string is a random string of length n, and then bound the probability that some algorithm reconstructs the target string. We will use $A = a_1 \cdots a_n$ denote the target string. In the following, we say that an event happens with high probability (w.h.p.) if its probability is $1 - n^{-\Omega(1)}$.

Our algorithms have the same basic structure:

```
1. $k \leftarrow k_0$.
2. Let $Q = \Sigma^k$. Ask the queries in $Q$ and construct $\mathrm{SP}^{A,k}$.
3. For $t = 1, \ldots, T$ do:
   (a) $\mathrm{SP}^{A,k+k_t} \leftarrow \mathrm{Extend}(\mathrm{SP}^{A,k}, k_t)$.
   (b) $k \leftarrow k + k_t$.
4. Reconstruct the string from $\mathrm{SP}^{A,k}$.
```

Procedure Extend uses $\mathrm{SP}^{A,k}$ and one round of queries in order to build $\mathrm{SP}^{A,k+k_t}$ (implementations of Extend will be given in Sections 3 and 4). If at step 4 of the algorithm the value of k is $2 \log n + s$, then A will be reconstructed correctly with probability $1 - O(4^{-s})$ [15]. In particular, if $s = \Omega(\log n)$ then A will be reconstructed correctly with high probability. Our goal in the next sections is to design procedure Extend, analyze its performance, and choose the parameters $k_0, \ldots, k_T$.

The following theorem will be used to bound the number of queries.

Theorem 1 (McDiarmid's bound [13]). *Let $f \colon \mathbb{R}^n \to \mathbb{R}$ be a function such that $|f(x) - f(x')| \leq c_i$ if x and x' differ only on the i-th coordinate. Let $Z_1, \ldots, Z_n$ be independent random variables. Then,*

$$\mathrm{P}\left[f(Z_1, \ldots, Z_n) - \mathrm{E}\left[f(Z_1, \ldots, Z_n)\right] > t\right] \leq \exp\left(\frac{-t^2}{\sum_{i=1}^n c_i^2}\right).$$

3 Binary model

In this section, we consider the binary model. Procedure Extend($\mathrm{SP}^{A,k}, \Delta$) is as follows:

1. Let Q_A be the set of all strings $s_1 \cdots s_{k+\Delta}$ such that $\mathrm{SP}^{A,k}(s_i \cdots s_{i+k-1}) = 1$ for all $i \in \{1, \dots, \Delta\}$.
2. Ask the queries in Q_A.
3. For every string B of length $k+\Delta$, set $\mathrm{SP}^{A,k+\Delta}(B) = 1$ if $B \in Q_A$ and the answer for B was 'yes', and set $\mathrm{SP}^{A,k+\Delta}(B) = 0$ otherwise.

We give a small example of procedure Extend: Let $A =$ CGGATGAG, $k = 3$, and $\Delta = 2$. The set Q_A contains all the substrings of A of length 5 (CGGAT, GGATG, GATGA, and ATGAG). Furthermore, Q_A contains the string CGGAG as all its substrings of length 3 (CGG, GGA, GAG) are substrings of A, and the strings ATGAT and TGATG.

The correctness of procedure Extend is trivial. We now estimate the number of queries that are asks by the procedure. Clearly, the number of queries in Q_A for which the answer is 'yes' is at most $n-(k+\Delta)+1$. It remains to bound the number of queries for which the answer is 'no'. Denote this number by Y.

Lemma 2. *If* $\Delta \leq k$ *then* $\mathrm{E}[Y] = O((k^2 4^{\Delta-k} + (n/4^k)k^3 4^{\Delta-k} + (n\Delta/4^k)e^{n\Delta/4^{k-1}}) \cdot n)$.

Proof. Let $\mathcal{B}$ be the set of all strings of length $k+\Delta$ that do not appear in A. One can compute $\mathrm{E}[Y]$ by computing $\mathrm{P}_{B\in\mathcal{B}}[B \in Q_A]$ (note that the probability $\mathrm{P}_{B\in\mathcal{B}}[B \in Q_A]$ is over both the random choice of A and the random choice of a string B from $\mathcal{B}$), and then multiplying this probability by $|\mathcal{B}|$. This can be extended as follows: For a partition of $\mathcal{B}$ into disjoint sets $\mathcal{B}_0, \dots, \mathcal{B}_l$, we have that $\mathrm{E}[Y] = \sum_i |\mathcal{B}_i| \cdot \mathrm{P}_{B\in\mathcal{B}_i}[B \in Q_A]$. The advantage of the latter approach over the former one is that an appropriate choice of the sets $\mathcal{B}_0, \dots, \mathcal{B}_l$ can simplify the computation of the probabilities. Since we are interested in an upper bound on $\mathrm{E}[Y]$, instead of a partition of $\mathcal{B}$ we will take subsets $\mathcal{B}_0, \dots, \mathcal{B}_l$ of $\mathcal{B}$ whose union is $\mathcal{B}$. Then, $\mathrm{E}[Y] \leq \sum_i |\mathcal{B}_i| \cdot \mathrm{P}_{B\in\mathcal{B}_i}[B \in Q_A]$.

The subsets of $\mathcal{B}$ are defined as follows. First, note that if $B \in Q_A$ then the prefix of B of length k is a substring of A. Therefore, defining $\mathcal{B}_0$ to be the set of all strings of length $k+\Delta$ whose prefixes of length k are not substrings of A, we have that $\mathrm{P}_{B\in\mathcal{B}_0}[B \in Q_A] = 0$. Every string in $\mathcal{B} \setminus \mathcal{B}_0$ is of the form $a_t \cdots a_{t+k+\Delta-s-1} b_1 \cdots b_s$ for some $1 \leq s \leq \Delta$ and $t \leq n-k-\Delta+s+1$, where $b_1 \neq a_{t+k+\Delta-s}$ (we assume here that a_{n+1} is a special character that is not equal to a character in Σ). We thus define

$$\mathcal{B}_s = \{a_t \cdots a_{t+k+\Delta-s-1} b_1 \cdots b_s : t \leq n-k-\Delta+s+1, b_1 \neq a_{t+k+\Delta-s}\}.$$

Clearly, $|\mathcal{B}_s| \leq n \cdot 3 \cdot 4^{s-1}$. For the rest of the proof, we will give a bound on $\mathrm{P}_{B\in\mathcal{B}_s}[B \in Q_A]$ for some fixed s.

Let $B = a_t \cdots a_{t+k+\Delta-s-1} b_1 \cdots b_s$ be a random string from $\mathcal{B}_s$. From the definition of procedure Extend, we have that $B \in Q_A$ if and only if there are indices $r_{\Delta-s+2}, \dots, r_{\Delta+1}$

such that $B_i^k = A_{r_i}^k$ for all i (note that we always have $B_i^k = A_{t+i-1}^k$ for $i = 1, \ldots, \Delta - s + 1$). Since $\Delta \leq k$, we have that every substring B_i^k for $i \geq \Delta - s + 2$ contains the character b_1, and from the fact that $b_1 \neq a_{t+k+\Delta-s}$ we conclude that $r_i \neq t + i - 1$ for all $i \geq \Delta - s + 2$.

Suppose we consider some fixed $r_{\Delta-s+2}, \ldots, r_{\Delta+1}$, and we want to compute the probability that $B_i^k = A_{r_i}^k$ for all i. These equality events may not be independent:

Example 1. if $r_{i+1} = r_i + 1$ then $\mathrm{P}\left[B_{i+1}^k = A_{r_{i+1}}^k \middle| B_i^k = A_{r_i}^k\right] = 1/4$.

Example 2. If $r_i = r_{i+1} = r_{i+2}$ then $\mathrm{P}\left[B_{i+2}^k = A_{r_{i+2}}^k \middle| B_i^k = A_{r_i}^k \wedge B_{i+1}^k = A_{r_{i+1}}^k\right] = 1/4$.

We can eliminate the dependencies of the type shown in Example 1 by grouping together equality events. More precisely, we say that two indices r_i and r_j are *adjacent* if $r_j - r_i = j - i$. For two adjacent indices r_i and r_j with $i < j$, the events $B_i^k = A_{r_i}^k$ and $B_j^k = A_{r_j}^k$ happen if and only if $B_i^{k+j-i} = A_{r_i}^{k+j-i}$ (this follows from the fact that $j - i \leq \Delta - 1 \leq k - 1$). More generally, for each equivalence class of the adjacency relation there is a corresponding equality event between a substring of A and a substring of B. We can assume that the equivalence classes of the adjacency relation have simple structure: We say that the indices $r_{\Delta-s+2}, \ldots, r_{\Delta+1}$ are *simple* if there are integers $\Delta+2-s = c_1 < c_2 < \cdots < c_x < c_{x+1} = \Delta+2$ such that the indices $r_{c_i}, r_{c_i+1}, \ldots, r_{c_{i+1}-1}$ form an equivalence class for $i = 1, \ldots, x$.

Claim 3. $B \in Q_A$ if and only if there are simple indices $r_{\Delta-s+2}, \ldots, r_{\Delta+1}$ such that $B_i^k = A_{r_i}^k$ for all i.

Proof. Suppose that $B \in Q_A$, and let $r_{\Delta-s+2}, \ldots, r_{\Delta+1}$ be (not necessarily simple) indices such that $B_i^k = A_{r_i}^k$ for all i. If r_i and r_j are adjacent indices, with $i < j$, then $B_l^k = A_{r_i+l-i}^k$ for every $l = i, \ldots, j$. Therefore, for every $l = i+1, \ldots, j-1$, if $r_l \neq r_i + (l - i)$ we can change the value of r_l to $r_i + (l - i)$. By repeating this process, we obtain the desired simple indices.

The other direction of the claim is trivial. ■

To compute (or bound) $\mathrm{P}_{B \in \mathcal{B}_s}[B \in Q_A]$, consider some fixed simple indices $r_{\Delta-s+2}, \ldots, r_{\Delta+1}$. We want to compute the probability that $B_i^k = A_{r_i}^k$ for all i, or equivalently, the probability that $B_{c_i}^{k-1+c_{i+1}-c_i} = A_{r_{c_i}}^{k-1+c_{i+1}-c_i}$ for $i = 1, \ldots, x$. Each string $A_{r_{c_i}}^{k-1+c_{i+1}-c_i}$ is called a *block* and will be denoted by L_i. We also define block L_0 to be the string $A_t^{k+\Delta}$. The *starting position* in A of block L_i is r_{c_i}, and the starting position of L_i in B is c_i. Two blocks L_i and L_j *overlap* if their occurrences in A have common letters (in other words, for $i < j$, L_i and L_j overlap if $[r_{c_i}, r_{c_i} + |L_i| - 1] \cap [r_{c_j}, r_{c_j} + |L_j| - 1] \neq \emptyset$).

We consider three cases. The first case is when there are no overlapping blocks. In this case, the events $\{B_{c_i}^{k-1+c_{i+1}-c_i} = A_{r_{c_i}}^{k-1+c_{i+1}-c_i}\}_{i=1}^x$ are independent, so the probability that these events happen for fixed $r_{\Delta+2-s}, \ldots, r_{\Delta+1}$ is

$$\prod_{i=1}^{x} \frac{1}{4^{k-1+c_{i+1}-c_i}} = \frac{1}{4^{\sum_{i=1}^{x}(k-1+c_{i+1}-c_i)}} = \frac{1}{4^{(k-1)x+s}}.$$

For fixed x, the number of ways to choose simple indices $r_{\Delta+2-s}, \ldots, r_{\Delta+1}$ which have x equivalence classes is bounded by $\binom{s-1}{x-1}n^x$. Therefore, the contribution of the first case to $\mathrm{P}_{B \in \mathcal{B}_s}\left[B \in Q_A\right]$ is at most

$$\begin{aligned}\sum_{x=1}^{s}\binom{s-1}{x-1}\frac{n^x}{4^{(k-1)x+s}} &= \frac{n}{4^{k-1+s}}\sum_{x=1}^{s}\binom{s-1}{x-1}\left(\frac{n}{4^{k-1}}\right)^{x-1} \\ &= \frac{n}{4^{k-1+s}}\left(1+\frac{n}{4^{k-1}}\right)^{s-1} \leq \frac{n}{4^{k-1+s}} \cdot e^{n(s-1)/4^{k-1}}.\end{aligned}$$

In the next two cases, assume that there are overlapping blocks, and let L_i and L_j be two blocks that overlap with $i < j$. If $i > 0$ then consider the events $B^k_{c_i} = A^k_{r_{c_i}}$ and $B^k_{c_j} = A^k_{r_{c_j}}$. These two events are independent (see the proof of Lemma 6 in the next section), so the probability that these events happen (for fixed c_i, c_j, r_{c_i}, and r_{c_j}) is $1/4^{2k}$. The number of ways to choose c_i and c_j is $\binom{s}{2} \leq \Delta^2/2$, and the number of ways to choose r_{c_i} and r_{c_j} is at most $2(k+\Delta)n$ (as $|r_{c_i} - r_{c_j}| \leq k+\Delta-1$), so the contribution of the second case to $\mathrm{P}_{B \in \mathcal{B}_s}\left[B \in Q_A\right]$ is bounded by $(k+\Delta)\Delta^2 n/4^{2k} \leq 2k^3 n/4^{2k}$.

The last case is when $i = 0$. The event $B^k_{c_j} = A^k_{r_{c_j}}$ is composed of k equalities between the $c_j + l$-th letter of B and the $r_{c_j} + l$-th letter of A for $l = 0, \ldots, k-1$. Each such equality adds a requirement that either two letters of A are equal (if $i + j \leq k + \Delta - s$), or a letter in $b_1 \cdots b_s$ is equal to a letter in A. In either case, the probability that such equality happens given that the previous equalities happen is exactly $1/4$, as at least one of the two letters of the equality is not restricted by the previous equalities. Therefore, for fixed c_j and r_{c_j}, the probability that $B^k_{c_j} = A^k_{r_{c_j}}$ is $1/4^k$. The number of ways to choose c_j is $s \leq \Delta$, and the number of ways to choose r_{c_j} is at most $2(k+\Delta)$. Thus, the contribution of the first case to $\mathrm{P}_{B \in \mathcal{B}_s}\left[B \in Q_A\right]$ is at most $4k^2/4^k$.

Combining the three cases, we obtain that

$$\mathrm{P}_{B \in \mathcal{B}_s}\left[B \in Q_A\right] \leq \frac{4k^2}{4^k} + \frac{2k^3 n}{4^{2k}} + \frac{n}{4^{k-1+s}} \cdot e^{n\Delta/4^{k-1}}.$$

Therefore,

$$\begin{aligned}\mathrm{E}\left[Y\right] &\leq \sum_{s=1}^{\Delta} n \cdot 3 \cdot 4^{s-1} \cdot \mathrm{P}_{B \in \mathcal{B}_s}\left[B \in Q_A\right] \\ &\leq \left(4k^2 4^{\Delta-k} + 2\left(\frac{n}{4^k}\right)k^3 4^{\Delta-k} + 3\left(\frac{n\Delta}{4^k}\right)e^{n\Delta/4^{k-1}}\right) \cdot n. \qquad \blacksquare\end{aligned}$$

Lemma 4. *If $k \geq \log n$, $k = O(\log n)$, and $\Delta \leq 0.48 \cdot \log n$, then w.h.p., $Y = O((n\Delta/4^k)e^{n\Delta/4^{k-1}} \cdot n) + o(n)$.*

Proof. By Lemma 2,

$$\begin{aligned}\mathrm{E}\left[Y\right] &= O((\log^2 n + (n/4^k) \cdot \log^3 n) \cdot 4^{-0.52\log n} \cdot n + (n\Delta/4^k)e^{n\Delta/4^{k-1}} \cdot n) \\ &= O((n\Delta/4^k)e^{n\Delta/4^{k-1}} \cdot n) + o(n).\end{aligned}$$

The random variable Y is a function of the random variables $a_1, \dots, a_n$. A change in one letter a_i changes k substrings of A of length k. For a single k-substring of A, the number of strings of length $k+\Delta$ that contain it is at most $(\Delta+1)4^{\Delta}$. Therefore, a change in one letter of A changes the value of Y by at most $k(\Delta+1)4^{\Delta} = O(n^{0.48}\log^2 n)$. Using Theorem 1,

$$\mathrm{P}\left[Y - \mathrm{E}\left[Y\right] > n^{0.99}\right] \le \exp\left(\frac{-n^{2\cdot 0.99}}{n \cdot O\left(n^{0.48}\log^2 n\right)^2}\right) = e^{-\Omega(n^{0.02}/\log^4 n)},$$

and the lemma follows. ■

We are now ready to present our first algorithm, which will be called algorithm A. We first define f_i to be a tower of fours of height i (i.e., $f_1 = 4$ and $f_i = 4^{f_{i-1}}$ for $i > 1$). Algorithm A is based on the algorithm given in Section 2 with procedure Extend described in the beginning of this section, and with the following parameters: $T = \log^* n - 1$, $k_0 = \lceil \log n \rceil$, and $k_t = \min(f_{t+3}, 0.48 \cdot \log n)$ for $t = 1, \dots, T$.

Theorem 5. *With high probability, algorithm A reconstructs a random string of length n and uses $O(n)$ queries in each round.*

Proof. Since $f_{\log^* n - 1} > 0.48 \log n$, we get that $k_T = k_{T-1} = k_{T-2} = 0.48 \cdot \log n$. Thus, $\sum_{t=0}^{T} k_t > 2.1 \cdot \log n$, so the algorithm reconstructs the target string with high probability.

The number of queries in the first round is $4^{k_0} \le 4n$. Let $l_t = \sum_{i=0}^{t-1} k_i$ and $L_t = nk_t/4^{l_t - 1}$. We claim that $L_t \le L_1$ for all $t \ge 2$. The proof of this claim is simple as

$$L_t = \frac{nk_t}{4^{l_t - 1}} \le \frac{n4^{k_{t-1}}}{4^{l_t - 1}} = \frac{n}{4^{l_{t-1}-1}} \le L_{t-1}.$$

By Lemma 4, w.h.p., the number of queries in round t is $n + O(L_{t-1}e^{L_{t-1}} \cdot n) + o(n)$. Since $L_t \le L_1 \le nf_4/4^{k_0 - 1} = O(1)$, it follows that the number of queries in each round is $O(n)$. ■

4 Ternary model

For the ternary model, we use a procedure called Extend$_2$, that is based on the algorithm of Frieze and Halldórsson [8]:

1. Let Q_A be the set of all strings $s_1 \cdots s_{k'}$ of length $k' \in \{k+1, \dots, k+\Delta\}$ such that $\mathrm{SP}^{A,k}(s_1 \cdots s_k) \ge 1$, $\mathrm{SP}^{A,k}(s_{k'-k+1} \cdots s_k) \ge 1$, and $\mathrm{SP}^{A,k}(s_i \cdots s_{i+k-1}) = 2$ for $i = 2, \dots, k'-k$.

2. Ask the queries in Q_A and construct $\mathrm{SP}^{A,k+\Delta}$.

The correctness of procedure Extend follows from [8].

Lemma 6. *If $k \geq \log n + 2$ and $\Delta < k$, the expected number of queries asked by* $\text{Extend}_2(SP^{A,k}, \Delta)$ *is $O(n)$.*

Proof. The proof is similar to the proof of Lemma 2. We first bound the number of 'no' queries. For this purpose, define $\mathcal{B}$ to be the set of all strings of lengths between $k+1$ and $k+\Delta$ that do not appear in A. We define subsets of $\mathcal{B}$ as follows: First, let $\mathcal{B}_0$ to be the set of all strings in $\mathcal{B}$ whose prefixes of length k are not substrings of A. Next, for $s \in \{1, \ldots, \Delta\}$ and $l \in \{0, \ldots, \Delta - s\}$, let

$$\mathcal{B}_{s,l} = \{a_t \cdots a_{t+k+l-1} b_1 \cdots b_s : t \leq n - k - l + 1, b_1 \neq a_{t+k+l}\},$$

and we have $|\mathcal{B}_{s,l}| \leq n \cdot 3 \cdot 4^{s-1}$. Fix some s and l, and let $B = a_t \cdots a_{t+k+l-1} b_1 \cdots b_s$ be a random string from $\mathcal{B}_{s,l}$.

Claim 7. $B \in Q_A$ if and only if there are indices $r^1_{l+2}, \ldots, r^1_{l+s+1}$ and $r^2_2, \ldots, r^2_{l+s}$ such that

1. $B^k_i = A^k_{r^j_i}$ for all i and j.

2. $r^j_i \neq t + i - 1$ for all i and j.

3. $r^1_i \neq r^2_i$ for all i.

We say that the indices $r^1_{l+2}, \ldots, r^1_{l+s+1}, r^2_2, \ldots, r^2_{l+s}$ are *simple* if each equivalence classes of the adjacency relation is of the form $r^j_i, r^j_{i+1}, \ldots, r^j_{i'}$ (recall that two indices r^j_i and $r^{j'}_{i'}$ are adjacent if $r^j_i - r^{j'}_{i'} = i - i'$). Similarly to the proof of Claim 3, if $B \in Q_A$ then we can obtain indices that satisfy properties 1–3 of Claim 7 and are "almost" simple, where "almost" means that all the equivalence classes are of the form $r^j_i, r^j_{i+1}, \ldots, r^j_{i'}$, except perhaps the class that contains r^1_{l+s+1}. By ignoring r^1_{l+s+1}, we obtain the following:

Claim 8. If $B \in Q_A$ then there are simple indices $r^1_{l+2}, \ldots, r^1_{l+s}, r^2_2, \ldots, r^2_{l+s}$ that satisfy properties 1–3 of Claim 7.

Our goal is to give an upper bound on the probability that $B \in Q_A$. Using Claim 8, we look at some fixed simple indices $r^1_{l+2}, \ldots, r^1_{l+s}, r^2_2, \ldots, r^2_{l+s}$ that satisfy properties 2–3 of Claim 7, and we will bound the probability that $B^k_i = A^k_{r^j_i}$ for all i and j. We denote this event by $\mathcal{E}$.

Denote $R_1 = \{r^1_{l+2}, \ldots, r^1_{l+s}\}$ and $R_2 = \{r^2_2, \ldots, r^2_{l+s}\}$. Let $L^1_1, \ldots, L^1_{x_1}$ be the blocks corresponding to the indices in R_1, and let $L^2_1, \ldots, L^2_{x_2}$ be the blocks corresponding to the indices in R_2. As before, we define block L^1_0 to be the string $A^{k+\Delta}_t$. We say that two blocks are *far* if the distance between their starting positions in A is at least $6k$. An important property of this definition is that for a set $\mathcal{L}$ of pairwise far blocks, a block (not in $\mathcal{L}$) can overlap with at most one block from $\mathcal{L}$. Moreover, two blocks that each overlaps with a distinct block in $\mathcal{L}$ cannot overlap with each other.

To help with the proof of the lemma, we build a simple bipartite graph $G = (V, E)$. The set of vertices V contains vertices $v_1, \ldots, v_n$ that correspond to the n characters

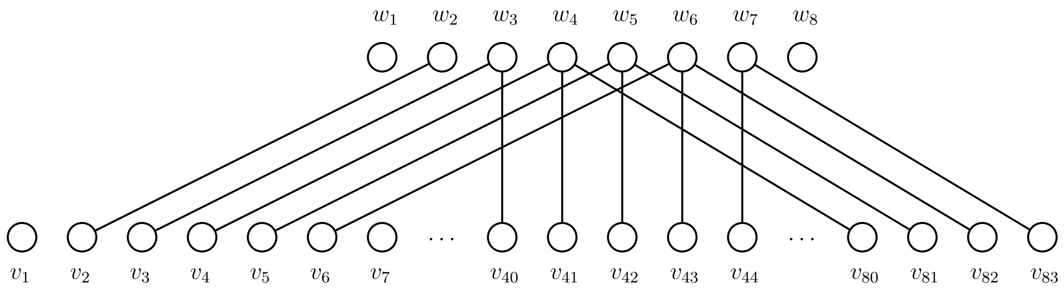

Figure 1: Example of the graph G for the case when the blocks are pairwise far (case 1). In this example $k = 4$, $l = 1$, and $s = 3$. The indices are $r_3^1 = 40$, $r_4^1 = 41$, $r_2^2 = 2$, $r_3^2 = 3$, and $r_4^2 = 80$. The equivalence classes are $\{r_3^1, r_4^1\}$, $\{r_2^2, r_3^2\}$, and $\{r_4^2\}$. All the edges of the graph are shown, except for the special edges.

of A, and vertices $w_1, \ldots, w_{k+l+s}$ that correspond to the $k + l + s$ characters of B. Each equality event $B_i^k = A_{r_i^j}^k$ consists of k letters equalities. The graph G contains k edges $(w_i, v_{r_i^j}), \ldots, (w_{i+k-1}, v_{r_i^j+k-1})$ that correspond to these equalities. Additionally, G contain *special edges* $(w_1, v_t), \ldots, (w_{k+l}, v_{t+k+l-1})$. The set of special edges is denoted by $\hat{E}$. See Figure 1 for an example. The probability that event $\mathcal{E}$ happens is $\prod_{C \in \mathrm{CC}(G)} 4^{-(|C|-\hat{e}(C)-1)}$, where $\mathrm{CC}(G)$ is the set of connected component of G, $|C|$ is number of vertices in the connected component C, and $\hat{e}(C)$ is the number of special edges in C. In particular, if G has no cycles, then $\mathrm{P}[\mathcal{E}] = 4^{-|E \setminus \hat{E}|}$. Moreover, for every set of edges E' with $\hat{E} \subseteq E' \subseteq E$ such that the graph (V, E') has no cycle, we have $\mathrm{P}[\mathcal{E}] \le 4^{-|E' \setminus \hat{E}|}$.

We consider several cases:

Case 1 The blocks are pairwise far. Since the graph G does not contain cycles in this case (see Figure 1), the probability that event $\mathcal{E}$ happens is $1/4^{(k-1)(x_1+x_2)+l+2s-2}$. For fixed x_1 and x_2, the number of ways to choose the indices $r_{l+2}^1, \ldots, r_{l+s}^1, r_2^2, \ldots, r_{l+s}^2$ is at most $\binom{(s-1)-1}{x_1-1}\binom{(l+s-1)-1}{x_2-1} n^{x_1+x_2}$. Therefore, the contribution of this case to $\mathrm{P}_{B \in \mathcal{B}_{s,l}}[B \in Q_A]$ is bounded by

$$\begin{aligned}
&\sum_{x_1=1}^{s-1} \sum_{x_2=1}^{l+s-1} \binom{(s-1)-1}{x_1-1} \binom{(l+s-1)-1}{x_2-1} n^{x_1+x_2} \frac{1}{4^{(k-1)(x_1+x_2)+l+2s-2}} \\
&= \frac{n^2}{4^{2(k-1)+l+2s-2}} \sum_{x_1=1}^{s-1} \binom{s-2}{x_1-1} \left(\frac{n}{4^{k-1}}\right)^{x_1-1} \sum_{x_2=1}^{l+s-1} \binom{l+s-2}{x_2-1} \left(\frac{n}{4^{k-1}}\right)^{x_2-1} \\
&= \frac{n^2}{4^{2k+l+2s-4}} \left(1 + \frac{n}{4^{k-1}}\right)^{l+2s-4} = O\left(\frac{n^2}{4^{2k+l+2s}} \cdot e^{2(l+s)n/4^{k-1}}\right) \\
&= O\left(\frac{1}{4^{l+2s}} \cdot e^{2(l+s)n/4^{k-1}}\right),
\end{aligned}$$

where the last equality follows from the fact that $k \ge \log n + 2$.

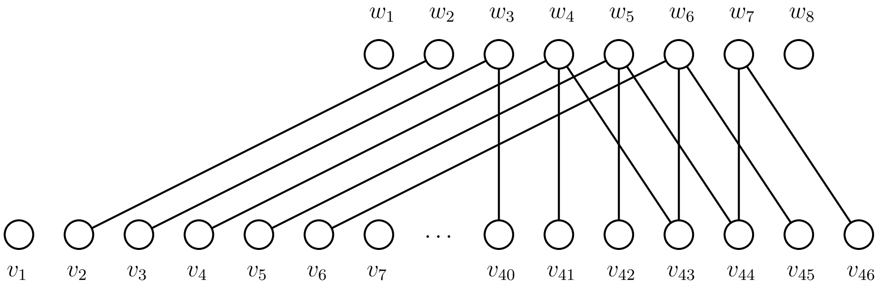

Figure 2: Example of the graph G in case 2.

We now assume that there are near blocks. We define a linear order $<$ on the blocks $L^1_0, L^1_1, \ldots, L^1_{x_1}, L^2_1, \ldots, L^2_{x_2}$ according to their starting positions in B with equalities broken arbitrarily (L^1_0 is the first block in the order). Let $L^i_{\alpha'}$ be the block such that

1. $L^i_{\alpha'}$ is near to a block that appears before $L^i_{\alpha'}$ in the order $<$.
2. $L^i_{\alpha'}$ is the first block in $<$ among all the blocks that satisfy 1.

Let $L^j_{\gamma'}$ be the block such that

1. $L^j_{\gamma'}$ is near to either $L^i_{\alpha'}$, a block that appears before $L^i_{\alpha'}$ in $<$, or a block that appear after $L^j_{\gamma'}$ in $<$.
2. $L^j_{\gamma'}$ is the last block in $<$ among all the blocks that satisfy 1.

Denote $\alpha = \alpha' - 1$ and $\gamma = x_j - \gamma'$. Let β be the number of indices from R_1 that correspond to the blocks $L^i_1, \ldots, L^i_\alpha$, and let δ be the number of indices from R_j that correspond to the blocks $L^j_{\gamma'+1}, \ldots, L^j_{x_j}$.

Case 2 The blocks $L^i_{\alpha'}$ and $L^j_{\gamma'}$ are the same block. This implies that in every pair of near blocks, one of the blocks is $L^i_{\alpha'}$ and the other block is a block that appears before $L^i_{\alpha'}$ in $<$. Since the blocks that appear before $L^i_{\alpha'}$ are pairwise far, it follows that there is at most on pair of overlapping blocks. It is easy to verify that in this case, the graph G has no cycles (see Figure 2). Thus, for fixed $r^1_{l+2}, \ldots, r^1_{l+s}, r^2_2, \ldots, r^2_{l+s}$, the probability of event $\mathcal{E}$ is the same as in case 1. The upper bound on the number of ways to choose $r^1_{l+2}, \ldots, r^1_{l+s}, r^2_2, \ldots, r^2_{l+s}$ for case 1 is also an upper bound on the number of ways to choose these indices for both case 1 and case 2. Therefore, contribution of case 2 to $\mathrm{P}_{B\in\mathcal{B}_{s,l}}[B \in Q_A]$ is accounted in case 1.

For the next cases assume that $L^i_{\alpha'}$ and $L^j_{\gamma'}$ are different blocks.

Case 3 $i = 1$, $\alpha > 0$ and $\gamma > 0$. Consider the following sets of edges in G:

1. E_1 = edges that correspond to the first β indices from R_1, first $l + \beta$ indices from R_2, last δ indices from R_1, and last δ indices from R_2.

2. $E_2 =$ edges that correspond to $r^1_{l+2+\beta}$.

3. $E_3 =$ edges that correspond to $r^j_{l+s-\delta}$.

Note that $r^1_{l+2+\beta}$ is the first index that corresponds to the block $L^i_{\alpha'}$, and $r^j_{l+s-\delta}$ is the last index that corresponds to the block $L^j_{\gamma'+1}$. The first β indices from R_1 form α equivalence classes (corresponding to the blocks $L^1_1, \ldots, L^1_\alpha$). Let α_2 be the number of equivalence classes in the first $l+\beta$ indices from R_2. Similarly, the last δ indices from R_j form γ equivalence classes, and define γ_2 to be the number of equivalence classes in the last δ indices from R_{3-j}.

Let $W_2 = \{w_{l+\beta+2}, \ldots, w_{k+l+\beta+1}\}$ (resp., $W_3 = \{w_{l+s-\delta}, \ldots, w_{k+l+s-\delta-1}\}$) be the set of vertices among $w_1, \ldots, w_{k+l+s}$ which are incident with an edge of E_2 (resp., E_3). Define E'_3 to be the set of edges from E_3 that are incident with a vertex from $W_3 \setminus W_2$.

Claim 9. The graph $G' = (V, \hat{E} \cup E_1 \cup E_2 \cup E'_3)$ has no cycles.

Proof. Let $\mathcal{L}$ be the set of blocks that appear before $L^i_{\alpha'}$ or after $L^j_{\gamma'}$ in $<$. The blocks in $\mathcal{L}$ are pairwise far. Therefore, the blocks $L^i_{\alpha'}$ and $L^j_{\gamma'}$ can each overlap with at most one block in $\mathcal{L}$. Thus, the subgraphs $G_2 = (V, \hat{E} \cup E_1 \cup E_2)$ and $G_3 = (V, \hat{E} \cup E_1 \cup E'_3)$ have no cycles. Assume first that $L^j_{\gamma'}$ does not overlap with a block that appears after $L^j_{\gamma'}$ in $<$. For each vertex from $W_3 \setminus W_2$, all its neighbors in G_2 have degree 1 (in G_2). Hence, adding the edges in E'_3 to G_2 cannot creates cycles, and we conclude that G' has no cycles.

Now assume that $L^j_{\gamma'}$ overlaps with a block that appears after $L^j_{\gamma'}$ in $<$. From the definition of near blocks, we have 3 possible cases:

1. $L^i_{\alpha'}$ overlaps with $L^j_{\gamma'}$, and does not overlap with any of the blocks in $\mathcal{L}$.

2. $L^i_{\alpha'}$ overlaps with with one block from $\mathcal{L}$ (that appear before $L^i_{\alpha'}$ in $<$), and does not overlap with $L^j_{\gamma'}$.

3. $L^i_{\alpha'}$ does not overlap with $L^j_{\gamma'}$ and does not overlap with any of the blocks in $\mathcal{L}$.

In the first case we have that for each vertex from W_2, all its neighbors in G_3 have degree 1. It follows that G' has no cycles. In the last two cases, we have that a connected component of G' that contains an edge from E'_3 cannot contain an edge from E_2. Therefore, G' has no cycles. ■

Since G' has no cycles, $\mathrm{P}[\mathcal{E}] \leq 1/4^{|E_1|+|E_2|+|E'_3|}$. We have

$$|E_1| = ((k-1)\alpha + \beta) + ((k-1)\alpha_2 + l + \beta) + ((k-1)\gamma + \delta) + ((k-1)\gamma_2 + \delta),$$
$$|E_2| = k,$$

and

$$|E_3| = s - \beta - \delta - 1.$$

For fixed α, α_2, β, γ, γ_2, and δ, the number of ways to choose the indices that correspond to the edges of E_1 is at most $\binom{\beta-1}{\alpha-1}\binom{l+\beta-1}{\alpha_2-1}\binom{\delta-1}{\gamma-1}\binom{\delta-1}{\gamma_2-1}n^{\alpha+\alpha_2+\gamma+\gamma_2}$. There are at most $(\alpha+\alpha_2+1)\cdot 8k \leq 16k^2$ ways to choose $r^1_{l+2+\beta}$ (since the block of $r^1_{l+2+\beta}$ must be near one of the blocks $L^1_0, L^1_1, \ldots, L^1_\alpha, L^2_1, \ldots, L^2_{\alpha_2}$), and at most $(\alpha+\alpha_2+\gamma+\gamma_2+2)\cdot 8k \leq 32k^2$ ways to choose $r^j_{l+s-\delta}$. Therefore, the contribution of this case to $\mathrm{P}_{B\in\mathcal{B}_{s,l}}[B \in Q_A]$ is bounded by

$$\begin{aligned}
&\frac{512k^4}{4^{k-1+l+s}}\sum_{\beta=1}^{s-3}\sum_{\delta=1}^{s-\beta-2}\frac{1}{4^{\beta+\delta}}\sum_{\alpha=1}^{\beta}\binom{\beta-1}{\alpha-1}\frac{n^\alpha}{4^{(k-1)\alpha}}\sum_{\alpha_2=1}^{l+\beta}\binom{l+\beta-1}{\alpha_2-1}\frac{n^{\alpha_2}}{4^{(k-1)\alpha_2}}\\
&\qquad\qquad\cdot\sum_{\gamma=1}^{\delta}\binom{\delta-1}{\gamma-1}\frac{n^\gamma}{4^{(k-1)\gamma}}\sum_{\gamma_2=1}^{\delta}\binom{\delta-1}{\gamma_2-1}\frac{n^{\gamma_2}}{4^{(k-1)\gamma_2}}\\
&\leq \frac{512k^4n^4}{4^{5(k-1)+l+s}}\sum_{\beta=1}^{s-3}\sum_{\delta=1}^{s-\beta-2}\frac{1}{4^{\beta+\delta}}e^{(l+2\beta+2\delta)n/4^{k-1}}\\
&= \frac{512k^4n^4}{4^{5(k-1)+l+s}}e^{ln/4^{k-1}}\sum_{\beta=1}^{s-3}\left(\frac{e^{2n/4^{k-1}}}{4}\right)^\beta\sum_{\delta=1}^{s-\beta-2}\left(\frac{e^{2n/4^{k-1}}}{4}\right)^\delta\\
&= O\left(\frac{k^4n^4}{4^{5k+l+s}}e^{ln/4^{k-1}}\right) = O\left(\frac{k^4}{4^{k+l+s}}e^{ln/4^{k-1}}\right)
\end{aligned}$$

(for the last two equalities we use the fact that $k \geq \log n + 2$).

The analysis of the remaining cases is similar, and we omit it. Table 1 summarizes the different cases and their contribution to $\mathrm{P}_{B\in\mathcal{B}_{s,l}}[B \in Q_A]$.

Summing all cases, we have

$$\mathrm{P}_{B\in\mathcal{B}_{s,l}}[B \in Q_A] = O\left(\frac{1}{4^{l+2s}}e^{2(l+s)n/4^{k-1}} + \frac{k^5}{4^{k+l+s}}e^{ln/4^{k-1}}\right),$$

and

$$\begin{aligned}
\mathrm{E}[Y] &\leq \sum_{s=1}^{\Delta}\sum_{l=0}^{\Delta-s} n\cdot 3\cdot 4^{s-1}\cdot \mathrm{P}_{B\in\mathcal{B}_{s,l}}[B \in Q_A]\\
&= O\left(\sum_{s=1}^{\Delta}\left(\frac{e^{2n/4^{k-1}}}{4}\right)^s\sum_{l=0}^{\Delta}\left(\frac{e^{2n/4^{k-1}}}{4}\right)^l + \frac{k^5}{4^k}\sum_{s=1}^{\Delta}\sum_{l=0}^{\Delta}\left(\frac{e^{n/4^{k-1}}}{4}\right)^l\right)\\
&= O(1).
\end{aligned}$$

Using similar argument, the expected number of 'yes' queries is $O(n)$. ■

Algorithm B uses the following parameters: $T = 1$, $k_0 = \lceil \log n \rceil + 10$, and $k_1 = \lceil \log n \rceil$. By Lemma 6 and Markov's inequality, we obtain the following theorem.

Theorem 10. *With probability of at least* 0.99*, algorithm B reconstructs a random string of length* n *and uses* $O(n)$ *queries in each round.*

Case	Contribution
No near blocks or $L^i_{\alpha'} = L^j_{\gamma'}$	$O\left(\frac{1}{4^{l+2s}} e^{2(l+s)n/4^{k-1}}\right)$
$i = 1$, $\alpha > 0$, $\gamma > 0$ or $i = 2$, $\beta > l$, $\alpha > 0$, $\gamma > 0$	$O\left(\frac{k^4}{4^{k+l+s}} e^{ln/4^{k-1}}\right)$
$i = 1$, $\alpha = 0$, $\gamma > 0$ or $i = 2$, $\beta > l$, $\alpha = 0$, $\gamma > 0$	$O\left(\frac{k^4}{4^{k+l+s}} e^{ln/4^{k-1}}\right)$
$i = 1$, $\alpha > 0$, $\gamma = 0$ or $i = 2$, $\beta > l$, $\alpha > 0$, $\gamma = 0$	$O\left(\frac{k^4}{4^{k+l+s}} e^{ln/4^{k-1}}\right)$
$j = 1$, $\alpha = 0$, $\gamma = 0$ or $i = 2$, $\beta > l$, $\alpha = 0$, $\gamma = 0$	$O\left(\frac{k^4 n}{4^{2k+l+s}} e^{ln/4^{k-1}}\right)$
$i = 2$, $\alpha > 0$, $\gamma > 0$	$O\left(\frac{k^5}{4^{k+l+s}} e^{ln/4^{k-1}}\right)$
$i = 2$, $\beta \leq l$, $\alpha = 0$, $\gamma > 0$	$O\left(\frac{k^3}{4^{k+l+s}}\right)$
$i = 2$, $\beta \leq l$, $\alpha > 0$, $\gamma = 0$	$O\left(\frac{k^5}{4^{k+l+s}} e^{ln/4^{k-1}}\right)$
$i = 2$, $\beta \leq l$, $\alpha = 0$, $\gamma = 0$	$O\left(\frac{k^2}{4^{k+l+s}}\right)$

Table 1: Contribution of all cases to $\mathrm{P}_{B \in \mathcal{B}_{s,l}}\left[B \in Q_A\right]$.

5 Concluding remarks and open problems

For the string reconstruction problem with linear sized rounds, we have shown an $\log^*_\sigma n$ rounds algorithm for the binary model, and a 2 rounds algorithm for the ternary model. While our result for the ternary model is optimal, it remains an open problem to determine the minimum number of rounds required in the binary model.

References

[1] 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.

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

[3] 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.

[4] 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.

[5] 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.

[6] 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.

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

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

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

[10] S. A. Heath, F. P. Preparata, and J. Young. Sequencing by hybridization using direct and reverse cooperating spectra. *J. of Computational Biology*, 10(3/4):499–508, 2003.

[11] 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.

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

[13] C. McDiarmid. On the method of bounded differences. In *Surveys in Combinatorics*, London Math. Soc. Lectures Notes 141, pages 148–188. Cambridge Univ. Press, 1989.

[14] 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.

[15] P. A. Pevzner, Y. 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.

[16] F. P. Preparata. Sequencing-by-hybridization revisited: The analog-spectrum proposal. *IEEE/ACM Trans. on Computational Biology and Bioinformatics*, 1(1):46–52, 2004.

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

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

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

[20] D. Tsur. Tight bounds for string reconstruction using substring queries. In *Proc. 9th International Workshop on Randomization and Computation (RANDOM)*, LNCS 3624, pages 448–459, 2005.

[21] D. Tsur. Optimal probing patterns for sequencing by hybridization. In *Proc. 6th Workshop on Algorithms in Bioinformatics (WABI)*, pages 366–375, 2006.