Handling Long Targets and Errors in Sequencing by Hybridization

Eran Halperin * Shay Halperin † Tzvika Hartman ‡ Ron Shamir §

February 26, 2003

Abstract

Sequencing by hybridization (SBH) is a DNA sequencing technique, in which the sequence is reconstructed using its k-mer content. This content, which is called the *spectrum* of the sequence, is obtained by hybridization to a universal DNA array. Standard universal arrays contain all k-mers for some fixed k, typically 8 to 10. Currently, in spite of its promise and elegance, SBH is not competitive with standard gel-based sequencing methods. This is due to two main reasons: lack of tools to handle realistic levels of hybridization errors, and an inherent limitation on the length of uniquely reconstructible sequence by standard universal arrays.

In this paper we deal with both problems. We introduce a simple polynomial reconstruction algorithm which can be applied to spectra from standard arrays and has provable performance in the presence of both false negative and false positive errors. We also propose a novel design of chips containing universal bases, that differs from the one proposed by Preparata et al. We give a simple algorithm that uses spectra from such chips to reconstruct with high probability random sequences of length lower only by a squared

*CS Division, Soda Hall, University of California Berkeley, CA 94720-1776. E-mail: `eran@EECS.berkeley.EDU`.

†School of Computer Science, Tel-Aviv University, Tel-Aviv 69978, Israel. E-mail: `hshay@post.tau.ac.il`.

‡Corresponding author. Department of Applied Mathematics and Computer Science, Weizmann Institute, Rehovot 76100, Israel. E-mail: `tzvi@wisdom.weizmann.ac.il`.Phone: +972-8-934-3586.

§School of Computer Science, Tel-Aviv University, Tel-Aviv 69978, Israel. E-mail: `rshamir@post.tau.ac.il`. Phone: +972-3-640-5383. Fax: +972-3-640-5384.

log factor compared to the information theoretic bound. Our algorithm is very robust to errors, and has a provable performance even if there are both false negative and false positive errors. Simulations indicate that its sensitivity to errors is also very small in practice.

Keywords: sequencing by hybridization, DNA sequencing with errors, DNA arrays, universal bases.

1 Introduction

Sequencing by Hybridization (SBH) was proposed in the late eighties [3, 9] as an alternative approach to DNA sequencing. In this technique, a large set of short oligonucleotides (called *probes*) is arrayed on a solid surface, forming a DNA chip. A solution with copies of the target DNA sequence (the sequence we would like to read) is brought in contact with the chip. Oligonucleotides on the chip hybridize with reverse complement segments of the target DNA molecules.Using the hybridization signals, the subset of probes that hybridize with the target (the *spectrum* of the sequence) is detected, and a combinatorial algorithm reconstructs the DNA sequence from its spectrum. For an overview on SBH see [20, 21, 27].

Initially, SBH was designed for chips with probes consisting of all 4^k strings of length k over the DNA alphabet (A,C,G,T) (we call this design *classical SBH*). Assuming that the spectrum is error-free and that the multiplicity of each k-mer in the spectrum is known, the problem is reduced to finding an Eulerian path in a particular directed graph [19] and therefore is tractable. However, these two assumptions are not practical.

Real hybridizations are error prone. They contain both *false negative* errors (i.e., probes that match the sequence but are missing from the spectrum), as well as *false positives* (probes in the experimental spectrum that do not match any position in the sequence). For the case of false negatives only, several algorithms were proposed [7, 19]. When there are also false positive errors, there are three known algorithms. Lipshutz [14] provides an iterative algorithm which assumes some knowledge of the error rates, and assumes an error model in which false positives can be obtained only by mismatches of a base at one end of the k-mer. The algorithm may not always converge, but simulations indicate that this happens with low probability. Blazewicz et el. [6, 5] give two algorithms that minimize the number of errors and have worst case exponential running time. None of the algorithms that was previously proposed to address both types of errors in classical SBH has a theoretical proof that with high probability, the obtained sequence is the original one.

Another main obstacle is that often reconstruction is not unique. Theoretical studies [2, 10, 28] proved that the expected length of unambiguously reconstructible sequences by classical SBH with probes of length k is $O(2^k)$. This was also observed empirically [21, 22]. In contrast, a simple information-theoretic argument [25] yields an upper bound of $O(4^k)$ if there are no restrictions on the set of probes, that is, the length of the target sequence may be no more than linearly proportional to the number of probes on the DNA chip.

Several approaches were proposed to decrease the probability of ambiguous reconstruction. Among them are alternative chip designs [22, 25, 26], interactive protocols [11, 29, 24], using additional positional information [4, 13, 28], using a homologous reference sequence [18] and resolving ambiguities by restriction enzymes [30]. An important breakthrough result due to Preparata et al. [25, 26] used *universal bases*, i.e., bases which hybridize with each of the four standard DNA bases (Probes containing universal bases are also called *gapped*, as the universal bases form gaps of unspecified positions between specified ones). Preparata et al. have given a new chip design based on probes which contain universal bases and is provably optimal (i.e., achieves the information theoretic bound) up to a constant factor (all results here and below hold with high probability for random sequences of independent, equiprobable bases).

In this paper we deal with the problem of noise in SBH, as well as with the problem of ambiguous reconstruction. We first provide a polynomial algorithm which solves the classical SBH problem with both false negative and false positive errors. Our algorithm does not assume knowledge of the spectrum multiplicities. If the false positive rate is small and the probability of false negative is q, the probability that our algorithm fails to reconstruct a random sequence of length $O(2^{(1-3q)k})$ is bounded by an arbitrarily small constant. To the best of our knowledge, this is the first rigorous theoretical analysis of a polynomial algorithm for SBH with positive and negative errors. This answers the challenge posed by Pevzner and Waterman in [23, Problem 35].

We also provide an alternative chip design to the one of Preparata et al. [25, 26] which uses universal bases. In contrast to the latter design, which has a deterministic periodic structure, we use *randomized probes*, in which the locations of the real (specified) bases among the universal ones are chosen randomly. We provide a second polynomial algorithm to reconstruct the target from such spectrum. Here too we do not assume that the multiplicities are known. The probability that our algorithm fails to reconstruct a random sequence of length $O(\frac{4^k}{k})$, using $\Theta(k4^k)$ probes, is bounded by an arbitrarily small constant, assuming there are no hybridization errors. Hence, our design is optimal up to a squared log factor.

The main advantage of our randomized probes design is that the same algorithm has provable performance in the presence of errors. If the false positive rate is small and the probability of false negative is q, the probability that our algorithm fails to reconstruct a random sequence of length $O((1-q)\frac{4^k}{k})$, using $\Theta(\frac{k4^k}{1-q})$ probes, is bounded by an arbitrarily small constant. This is the first rigorous theoretical analysis of any algorithm for SBH with gapped probes in the presence of errors: The analyses of [25, 26] assumes error-free data. Doi and Imai [8] report on an empirical study on an extension of the algorithms of [25, 26] which handles errors. Furthermore, our simulations show that our algorithm is much more resistant to errors than both the algorithms of Preparata et al. and of [8].

The paper is organized as follows. In Section 2 we introduce the model, some definitions and useful lemmas. In Section 3 we describe and analyze the algorithm for classical SBH arrays. In Section 4 we introduce the new array design that uses universal bases and analyze its reconstruction algorithm. In Section 5 we describe our empirical study, and we conclude with some possible extensions in Section 6. A preliminary version of this paper has appeared in [12].

2 Preliminaries

In this section we give some basic definitions, describe our model, and state some known lemmas that we are going to use in the analysis.

Let $\Sigma = \{A, G, C, T\}$, where A, G, C and T represent (specified) nucleotide bases. Following [25, 26], we add a "don't care" symbol N, implemented using a universal base [15] (biologically, a universal base can hybridize to any base). The DNA sequence that we wish to reconstruct, also called the *target*, consists of specified bases only. A *probe* is a short sequence of universal and specified bases. A sequence that contains universal bases will sometimes be referred to as the gapped subsequence of its specified locations. Hence, AANNTNC is the same as the subsequence AA**T*C.

Definition 2.1 (Matching Probe) *Probe a* matches *sequence S if it appears contiguously in S.*

For example, a =AANNTNC matches TAAGCTGCC. Note that the probe must be fully contained in S, so a does not match AACGTA.

A DNA array is a set of probes (typically of equal length). For a given array, the *theoretical spectrum* of a target is the set of all the array probes that match the target. Note that we ignore multiplicities in the spectrum, so spectra are viewed as sets and not as multisets.

Definition 2.2 (Fooling Probe) *For a sequence S, which is not a subsequence of the target, a probe is called a* fooling probe *if it matches both S and the target.*

The (experimental) *spectrum* $\mathcal{ES}$ of a target is the set of probes in the array that were recorded by the experiment as present in the target. For errorless data, the theoretical and experimental spectra coincide. We model noise in the SBH experiment by adding some false positives and some false negatives to the theoretical spectrum $\mathcal{T}$: For every probe $a \in A$ independently, if $a \notin \mathcal{T}$ then $a \in \mathcal{ES}$ (i.e., a appears in the spectrum) with probability p. This is the false positive probability. If $a \in \mathcal{T}$ then a appears in the spectrum $\mathcal{ES}$ with probability $1-q$, so q is the false negative probability. This probability is independent of the number of locations that a matches. Since the probability that a probe matches more than one location is very low, this model approximates quite well a model which assumes that false negative errors occur independently at each location. We assume that p is small enough so that the number of probes in the spectrum is dominated by the probes that do belong to $\mathcal{T}$.

Definition 2.3 (Supporting Probe) *A probe a* supports *a sequence S if it matches S and appears in the spectrum.*

Any probe that is a subsequence of the target is by definition a matching probe and, in case there are no false negative errors, also a supporting probe. For example, given errorless k-mer spectrum from a classical array, there will typically be k supporting probes for any subsequence of the target of length $2k-1$ (if the subsequence is degenerate there may be fewer). When errors are introduced, not all matching probes support a true subsequence, due to false negatives. A sequence which is not a subsequence of the target may be supported by fooling probes (which are not false negatives) and by false positive probes (that do not appear in the theoretical spectrum). Note that the latter are not fooling probes.

The *SBH problem* is to reconstruct the target S from the spectrum $\mathcal{ES}$. We assume we are given a prefix of S, of length $i-1$, where i is the length of the probes. Note that there are biochemical methods to fulfill the prefix requirement (see [25]). We assume that S is chosen uniformly from all the DNA strings of a determined length m, i.e., each symbol is chosen uniformly and independently.

In our analysis, we will use the following Chernoff-like upper bounds for large deviations (their proof can be found, e.g., in [1, Appendix A]):

Lemma 2.4 *Let Z be a random variable with a binomial distribution $Z \sim B(n,p)$. For every $\lambda > 1$, $Pr(Z > \lambda pn) < \left(e^{\lambda-1}\lambda^{-\lambda}\right)^{pn}$.*

Lemma 2.5 *Let Z be a random variable with the binomial distribution, $Z \sim B(n,p)$. For every $a > 0$, $Pr[Z - np < -a] < e^{-a^2/2pn}$.*

3 Classical SBH with Errors

In this section we assume that the array consists of all 4^k possible sequences of length $k \geq 6$, over the alphabet $\Sigma = \{A, G, C, T\}$. When there are neither false positives nor false negatives, it was shown [10, 2, 28] that with high probability, the longest possible sequences to be uniquely reconstructed, are of length $O(2^k)$. In this section, we address the errors-prone case. We describe a simple algorithm, and show that it reconstructs (with high probability) sequences of length $\Theta(2^{k(1-3q)})$, for $q < \frac{1}{3}$. Notice that in the error-free case, when $q = 0$, our algorithm is optimal up to a constant. As the number of probes is 4^k, and the length of the sequence is $O(2^k)$, we assume that $p < \frac{1}{2^k}$. Note that the length of the reconstructed sequences is not affected by p.

Let the input sequence be $s_1, \ldots, s_m$. Suppose that we already know the sequence $s_1, \ldots, s_i$, and we want to find s_{i+1}. The reconstruction algorithm is described in Figure 1.

1. Enumerate all 4^k sequences $a = a_1, \ldots, a_k$.
2. Pick a sequence a' such that the number of probes supporting $s_{i-k+2}, \ldots, s_i, a'_1, \ldots, a'_k$ is maximal (breaking ties arbitrarily).
3. Set $s_{i+1} = a'_1$.

Figure 1: Algorithm A for Classical SBH.

The running time of algorithm A is $O(mk4^k)$. Each of the $(2k-1)$-long sequences tried will be called a *path*.

3.1 Analysis of the Algorithm

Theorem 3.1 *The probability that algorithm A fails is bounded by an arbitrary small constant, if* $m = O(2^{(1-3q)k})$.

Proof: At each step, the algorithm tries to extend the current sequence by all possible sequences of length k. A sequence path $s_{i-k+2}, \ldots, s_i, a'_1, \ldots, a'_k$ in which the first new base (corresponding to a'_1 in algorithm A) is wrong, will be called a *bad path*. Let us fix a possible bad path. Denote by P_{bad} the probability that the number of supporting probes of this bad path is greater or equal to the the number of probes which support the correct path (and hence in this case the algorithm fails). There are $\frac{3}{4}4^k$ possible bad paths, and the number of possible locations that might lead to an error is bounded by m. We have to show that the failure probability is bounded by a constant $\epsilon < 1$. Hence, what we need to prove is that

$$P_{bad} \cdot \frac{3}{4}4^k \cdot m \leq \epsilon$$

Let X be a random variable that counts the number of supporting probes for the correct path, and let Y be the number of supporting probes for the bad path. Clearly, $Y = Y_1 + Y_2$, where Y_1 is the number of supporting probes for the bad path that are fooling (probes that appear also in the theoretical spectrum), and Y_2 is the number of supporting probes arising from false positives. The algorithm fails if at some point $X \leq Y$.

Suppose there were i fooling probes for the bad path. It is easy to see that $X \sim B(k, 1-q)$, $Y_1 \sim B(i, 1-q)$, and $Y_2 \sim B(k-i, p)$. Note that these random variables depend on i. However, the value of i will be clear from the context, so the random variables are not written as functions of i.

We first prove the following lemma:

Lemma 3.2 *The probability that a bad path has i fooling probes is at most* $2(k-i+1) \cdot m \cdot 4^{-k-i}$.

Proof: In the bad path, the first extension base is incorrect and therefore, all i fooling probes match other locations in the sequence. (Note that for a false path that starts with $1 \leq j \leq k$ correct bases - and hence is not called a bad path - the first j probes that support such path match also the correct path. Thus, they appear in the theoretical spectrum even if they do not match other locations in the sequence.)

Consider first the case where the i fooling probes of the bad path all match a single location (with appropriate shifts), which means that $k-1+i$ consecutive symbols of the bad path appear consecutively elsewhere in the sequence. This happens with probability at most $(k-i+1) \cdot \frac{m}{4^{k-1+i}}$ (there are $k-i$ possible sequences of $k-1+i$ consecutive symbols in the bad path).

The more general case is that the fooling probes match $x \geq 1$ different locations. In this case there are x different subsequences of the bad path of lengths $k-1+i_1, \ldots, k-1+i_x$, $\sum_{j=1}^{x} i_j = i$ that appear elsewhere in the sequence. There are $\binom{i-1}{x-1}$ possibilities of such decompositions of i into i_j's (the same as the number of possibilities of choosing a multiset of $i-x$ elements out of x elements). There are $kx+i$ restricted symbols, at most $\binom{m}{x}$ possible sets of locations in the sequence and at most $(k-i+x)^x$ possible sets of indices in the bad path. (Note that the fact that the occurrences of the probes in the sequence may overlap does not affect the analysis, since the number of restricted symbols remains the same.) Therefore the probability that there are i fooling probes is at most

$$\sum_{x=1}^{i} \binom{i-1}{x-1} \binom{m}{x} (k-i+x)^x \, 4^{-kx-i} \leq$$

$$\frac{4^{-i}}{i} \sum_{x=1}^{i} \left(i(k-i+x) \cdot m \cdot 4^{-k}\right)^x \leq 2(k-i+1) \cdot m \cdot 4^{-k-i}$$

The last inequality is obtained since for $k \geq 6$, the first summand (the expression for $x = 1$) is bounded by $\frac{1}{2}$ (note that $x \leq i$), so the whole sum is bounded by twice the first summand. ■

Summing over the possible values of i, and using Lemma 3.2, we obtain

$$\begin{aligned} P_{bad} &= \sum_{i=0}^{k} Pr[X \leq Y] \cdot Pr[i \ fooling] \\ &\leq \sum_{i=0}^{k} Pr[X \leq Y] \cdot 2(k-i+1)m \cdot 4^{-k-i} \end{aligned}$$

By denoting $f(i) \stackrel{\text{def}}{=} Pr[X \leq Y] \cdot 2(k-i+1)m^2 \cdot 4^{-i}$ we get

$$m \cdot 4^k \cdot P_{bad} = \sum_{i=0}^{(1-2q)k} f(i) + \sum_{i=(1-2q)k+1}^{k} f(i) \tag{1}$$

which has to be bounded by $\frac{4}{3}\epsilon$.

We will bound separately the first and the second sum. Roughly speaking, the second sum is bounded using the fact that the number of fooling probes i is high, an event which happens with very low probability. The first sum is bounded by observing that when i is small, it is unlikely that the number of probes that support the correct path is close to i, since it is far from the expectation.

Let $\rho = m2^{-(1-3q)k}$. We will first bound the second sum in (1). In order to bound $Pr[X \leq Y]$ we only use the fact that $Pr[X \leq Y] \leq \frac{1}{2}$, and we get

$$\begin{aligned} \sum_{i=(1-2q)k+1}^{k} f(i) \leq 2(qk) \max_{i>(1-2q)k} f(i) \leq \\ 4(qk)^2 m^2 \cdot 4^{-k(1-2q)} \leq \rho^2, \end{aligned} \tag{2}$$

where the last inequality holds since for every integer $x \geq 0$, $2x^2 \leq 4^x$, and therefore, $4(qk)^2 \leq 4^{qk}$.

In order to bound the first sum in (1), we will first fix $i \leq (1-2q)k$, and bound $f(i)$. For every $j > i$,

$$Pr[X \leq Y] \leq Pr[X \leq j] + Pr[Y \geq j]$$

So our goal is to find a number j, such that $Pr[X \leq j] + Pr[Y \geq j]$ is very small.

We will first bound $Pr[X \leq j]$. For simplicity of notation, let $x = \frac{k-i}{kq}$, $y = \frac{k-j}{kq}$, and $\delta = 1 - \frac{y}{x}$, and thus, $i = k - kqx, j = k - kqy$, and $y = x(1-\delta)$. By applying Lemma 2.4 for $Z = k - X$, and for $\lambda = y$, we have

$$
\begin{aligned}
Pr[X \le j] &= Pr[k - X \ge yqk] \\
&< \left(\frac{e^{y-1}}{y^y}\right)^{qk} = \left(\frac{1}{e}\left(\frac{e}{y}\right)^y\right)^{qk}
\end{aligned} \tag{3}
$$

In order to bound $Pr[Y \ge j]$, we first note that $Y_1 \le i$ is always true, and thus it is sufficient to bound the probability $Pr(Y_2 \ge j - i)$. We can now apply again Lemma 2.4 for Y_2 with $\lambda = \frac{j-i}{(k-i)p} = \frac{\delta}{p}$, and get that

$$
\begin{aligned}
Pr[Y \ge j] &\le Pr[Y_2 \ge j - i] \\
&< \left(\frac{e^{(\delta-p)/p} p^{\delta/p}}{\delta^{\delta/p}}\right)^{xqkp} < \left(\frac{ep}{\delta}\right)^{\delta xqk}
\end{aligned} \tag{4}
$$

Denote by $\gamma = qk$. Define

$$g(x, \gamma) \stackrel{\text{def}}{=} 2(k - i + 1)m^2 4^{-i} Pr[X \le j]$$

and

$$h(x, \gamma) \stackrel{\text{def}}{=} 2(k - i + 1)m^2 4^{-i} Pr[Y \ge j],$$

such that $f(i) \le g(x, \gamma) + h(x, \gamma)$.

- From equation (3), we get that

$$
\begin{aligned}
g(x, \gamma) &= 2(k - i + 1)m^2 4^{-i} Pr[X \le j] \\
&= 2xqk\rho^2 4^{(x-3)qk} Pr[X \le j] \\
&< 2\rho^2 x\gamma \left(\frac{4^{x-3}}{e}\left(\frac{e}{y}\right)^y\right)^{\gamma} \\
&\le 2\rho^2 x \left(0.54 \cdot 4^{x-3}\left(\frac{e}{y}\right)^y\right)^{\gamma},
\end{aligned}
$$

  where the last inequality holds since $\gamma^{1/\gamma} \le 1.45$. Also, note that $g(x, \gamma)$ only depends on x and γ for given ϵ and ρ.

- From equation (4), we get

$$h(x, \gamma) = 2(k - i + 1)m^2 4^{-i} Pr[Y \ge j]$$

$$
\begin{aligned}
&= 2xqk\rho^2 4^{(x-3)qk} Pr[Y \geq j] \\
&\leq 2\rho^2 x\gamma \left(\frac{1}{64}\left(4\left(ep/\delta\right)^{\delta}\right)^{x}\right)^{\gamma} \\
&\leq 2\rho^2 x \left(\frac{1.45}{64}\left(4\left(ep/\delta\right)^{\delta}\right)^{x}\right)^{\gamma}
\end{aligned}
$$

When we set $\delta = 0.05$, we get

$$
\begin{aligned}
g(x,\gamma) &\leq 2\rho^2 x \left(\frac{0.54}{64}\right)^{\gamma} \left(\frac{11}{x^{0.95}}\right)^{x\gamma}, \\
h(x,\gamma) &\leq 2\rho^2 x \left(\frac{1.45}{64}\right)^{\gamma} \left(5p^{0.05}\right)^{x\gamma}
\end{aligned}
$$

We first assume that $\gamma \geq 1$. It is easy to verify that $g(x,\gamma) \leq 9\rho^2 0.95^{x\gamma}$, for $\gamma > 1, x \geq 2$, and that $h(x,\gamma) \leq \rho^2 0.95^{x\gamma}$ when $5p^{0.05} \leq 1$. As $p < \frac{1}{2^k}$, p is sufficiently small, for sufficiently large k (See Remark 3.3 regarding how to relax the restriction on p).

From the analysis above, we get that $f(i) < 10\rho^2 0.95^{k-i}$, and thus,

$$
\sum_{i \leq (1-2q)k} f(i) \leq 10\rho^2 \sum_{i=2}^{\infty} 0.95^i = 200\rho^2 \tag{5}
$$

Putting together equations (2) and (5), and letting $\rho = \sqrt{\frac{\epsilon}{151}}$, we get

$$
Pr[failure] \leq P_{bad} \cdot \frac{3}{4} 4^k \cdot m \leq \frac{3}{4} 201\rho^2 < \epsilon
$$

Hence, for every $0 < \epsilon < 1$, if $m \leq \sqrt{\frac{\epsilon}{151}} 2^{(1-3q)k}$ then the failure probability is bounded by ϵ.

The case where $\gamma < 1$ is trivial, as we can add artificial false negatives so that the false negative rate will be $\frac{1}{k}$, and then, we can recover a sequence of length $\rho 2^{k(1-3/k)} = \frac{\rho}{8} 2^k$. ■

Remark 3.3 We note that the restriction that $5p^{0.05} \leq 1$ can be relaxed, by compromising on other parameters. For example, one can verify that by using $\delta = 0.5$, we get that all we need in order to show that $h(x,\gamma)$ is small is that $p < 0.01$, and to bound $g(x,\gamma)$, it is sufficient to use $m = \rho \cdot 2^{k(1-12q)}$, and to use $x \geq 11$. For $x < 11$, we can bound the sum similarly to equation (2). Thus, in this way we get a shorter reconstructed sequence in terms of q, but we can tolerate a constant frequency of false positives.

Remark 3.4 In case there are only false negative errors, a similar analysis shows that the length of the sequence is asymptotically the same. However, the hidden constant is significantly smaller. It is clear from the proof of Theorem 3.1 that in the case that there are only false positives, p could be quite large, with a small constant factor loss in the length of the reconstructed sequence.

4 A Randomized Chip Design Using Universal Bases

In this section we describe a new array design, over the alphabet $\{A, G, C, T, N\}$, that is, this time we allow gapped probes. Preparata et al. [25, 26] described a set of $\Theta(4^k)$ gapped probes, and showed how a randomly chosen sequence of size $m = \Theta(4^k)$ can be determined unambiguously with high probability, in the absence of errors. We shall prove that our new design is noise resistant, i.e., false positives or false negatives have little affect on the length of constructible sequences.

We build *randomized* probes, and show that a sequence of length $m \approx \frac{4^k}{k}$ can be determined unambiguously, with high probability, using a simple algorithm, based on a spectrum of a set of $\Theta(k4^k)$ probes, in the presence of errors. Actually, we take $m = \alpha\frac{4^k}{\beta k}$ for some universal constant α and some constant β that depends on the error rates.

We choose a set of probes in the following way. Let c be a sufficiently large integer. The length of each probe is $c \cdot k + 1$. The last symbol of the probe is a *specified* base, that is, A, G, C or T. First we form βk subsets of probes $A_1, \ldots, A_{\beta k}$. For each subset, we pick a random set of k positions out of $\{1, \ldots, ck\}$ and form all possible 4^{k+1} probes with specified bases in the chosen positions and in the last one, and the rest universal. Our overall set of probes is the union of the above subsets. Thus, the total number of probes we use is $n = \beta k 4^{k+1}$.

We assume that we are given a prefix of length ck of the sequence. Let the input sequence be $s_1, \ldots, s_m$. Suppose that we already know the sequence $s_1, \ldots, s_i$, and we want to find s_{i+1}. The reconstruction algorithm is described in Figure 2.

As to the running time of algorithm B, at each step we have to check 4 probes from each of the βk subsets (one for each possible specified base in the probe's last position). Therefore, the total running time is $O(km)$.

1. For each specified base X, count how many probes support $s_{i-ck+1}, \ldots, s_i, X$.
2. Set s_{i+1} to be a base that has maximum support (breaking ties arbitrarily).

Figure 2: Algorithm B for SBH with universal bases.

4.1 Bounding Probability of Fooling Probes

The following definitions will be useful in the proofs of this section. We will extend definition 2.1: Probe *a partially matches* sequence S if its first ck bases (excluding the last specified base) appear contiguously in S. Probe *a matches* sequence S *in location* l if it appears contiguously in S, ending at location l.

Assume we are trying to extend the sequence at location l. In the absence of errors, a probe that supports a false base extension has to satisfy two conditions:

1. It partially matches S in location l.
2. It matches S in some other location l'.

Hence, it is a fooling probe. Notice that when errors are introduced, a supporting probe may be a false positive that does not satisfy the second condition.

Let S_x be the set of k indices of the specified bases of probe p_x, excluding the last specified base. We say that the two probes p_1 and p_2 *agree* if the values of the specified bases in indices $S_1 \cap S_2$ are the same in both probes. A probe $a = a_1, \ldots, a_{ck+1}$ Δ-*self-agrees* if for every position $1 \le j \le ck - \Delta$, $a_j = a_{j+\Delta}$ (using the convention that the universal base N equals to all bases). Note that the last specified base (at position $ck+1$) is not required to match the corresponding position. Note also that for $\Delta \ge ck$, every probe Δ-self-agrees.

Claim 4.1 *For each step of algorithm B, the probability that a false base will get one fooling probe is bounded by α.*

Proof: The fooling probe partially matches S at location l, and matches S in some other location l'. Let $\Delta = |l' - l|$. Obviously, a necessary condition is that it Δ-self-agrees.

We use the notation $S_x + y$ to denote the set S_x shifted by y. Let $i = |S_x \cap (S_x + \Delta)|$. In each one of the βk sets of probes, there are exactly 4^{k+1-i} probes which Δ-self-agree, since there are i constrained symbols. A probe is a fooling probe if it Δ-self-agrees, the $|S_x \cup (S_x + \Delta)| = 2k+1-i$ relevant specified bases match the sequence, and the base in location l' is different from the one in location l. The latter two events, happen with probability $\frac{3}{4}\frac{1}{4^{2k+1-i}}$.

We multiply this probability by the number of possible locations l' and by the total number of Δ-self-agreeing probes, and get that the probability that there is at least one fooling probe in the l^{th} location is

$$\frac{3}{4}m \cdot \frac{1}{4^{2k+1-i}} \cdot \beta k 4^{k+1-i} \leq \alpha$$

Notice that this probability is independent of i and β. ■

Claim 4.2 *The probability that a false base will get two fooling probes that arise from the same location l' is bounded by $\frac{exp(4k/(c-1))}{4^{4k}}$.*

Proof: Let the two fooling probes be p_1 and p_2. Assume first that $|l' - l| > ck$, and fix $i = |S_1 \cap S_2|$. Both probes partially match location l, and match location l'. Thus,

- p_1 and p_2 agree. This happens with probability $\frac{1}{4^i}$.
- The $|S_1 \cup S_2|$ specified bases of both probes p_1 and p_2 match the appropriate symbols in both locations l and l'. This happens with probability $\frac{1}{4^{2|S_1 \cup S_2|}} = \frac{1}{4^{4k-2i}}$.

Thus, the probability of this event, given $|S_1 \cap S_2| = i$, is $\frac{1}{4^{4k-i}}$.

For the case $|l' - l| \leq ck$, if p_1 and p_2 agree, we can consider a new "meta-probe" with specified bases in locations $S_1 \cup S_2$ with values induced naturally by the values of the specified bases of p_1 and p_2. By combining the above argument with the argument in the proof of claim 4.1, we get that in this case, the probability is $\frac{1}{4^{4k-2i}}$.

Now we will sum over all the values of i to get the desired probability. Notice first, that

$$Pr[|S_1 \cap S_2| = i] = \frac{\binom{k}{i}\binom{(c-1)k}{k-i}}{\binom{ck}{k}}. \tag{6}$$

Also notice that

$$\frac{\binom{k}{i}\binom{(c-1)k}{k-i}}{\binom{ck}{k}} \leq \frac{k^i}{i!}\frac{((c-1)k)^{k-i}}{(k-i)!}\frac{k!}{((c-1)k)^k} \leq \frac{k^i}{i!(c-1)^i}. \tag{7}$$

Now, by equations (6) and (7), we get

$$Pr[probes\ match] =$$

$$\sum_{i=0}^{k} Pr[probes\ match \mid |S_1 \cap S_2| = i] \cdot Pr[|S_1 \cap S_2| = i]$$

$$\leq \frac{1}{4^{4k}} \sum_{i=0}^{k} \frac{4^i k^i}{i!(c-1)^i}$$

$$\leq \frac{1}{4^{4k}} \sum_{i=0}^{\infty} \frac{1}{i!} (\frac{4k}{c-1})^i = \frac{exp(4k/(c-1))}{4^{4k}}$$

■

Claim 4.3 *For each step of algorithm B, the probability that a false base will get two fooling probes is bounded by* α^2*.*

Proof: Each fooling probe matches some location in the sequence. It could be the case that each probe matches a different location - that will happen, by Claim 4.1, with probability at most α^2, as these events are independent. Alternatively, they could arise from the same location l'. That is, both probes match the current location and also the same false location. There are at most m possible false locations and $\binom{n}{2}$ possible pairs of probes. Thus, using Claim 4.2, the probability of the above event is bounded by

$$mn^2 \frac{exp(4k/(c-1))}{4^{4k}} = \beta k \alpha \frac{exp(4k/(c-1))}{4^{k-2}} \leq \alpha^2,$$

for c large enough. ■

Claim 4.4 *The probability that throughout algorithm B, a false base will get three fooling probes from the same location in the sequence is exponentially small.*

Proof: Denote the three fooling probes by p_1, p_2, p_3. Assume $|S_1 \cap S_2| = i$ and $|(S_3 \cap (S_1 \cup S_2)| = j$. By the same arguments as in as in the proof of Claim 4.2, we get that the probability that the three probes agree is $\frac{1}{4^{i+j}}$, and the probability that they match both locations (given that they agree) is $\frac{1}{4^{2|S_1 \cup S_2 \cup S_3|}} = \frac{1}{4^{6k-2i-2j}}$.

Summing over all possible values of i and j, and using equations (6) and (7), we get

$$Pr[the\ 3\ probes\ match] =$$

$$\sum_{i=0}^{k} Pr[|S_1 \cap S_2| = i] \cdot$$

$$\sum_{j=0}^{k} Pr[|(S_3 \cap (S_1 \cup S_2)| = j||S_1 \cap S_2| = i] \cdot \frac{1}{4^{6k-i-j}}$$

$$\leq \frac{1}{4^{6k}} \sum_{i=0}^{k} \frac{1}{i!} (\frac{k}{c-1})^i \sum_{j=0}^{k} \frac{1}{j!} (\frac{2k-i}{c-2})^j \cdot 4^{i+j}$$

$$\leq \frac{1}{4^{6k}} \sum_{i=0}^{k} \frac{1}{i!} (\frac{4k}{c-1})^i \sum_{j=0}^{k} \frac{1}{j!} (\frac{8k}{c-2})^j$$

$$\leq \frac{1}{4^{6k}} \sum_{i=0}^{k} \frac{1}{i!} (\frac{4k}{c-1})^i exp(\frac{8k}{c-2})$$

$$\leq \frac{1}{4^{6k}} e^{4k/(c-1)} e^{8k/(c-2)} \approx \frac{e^{12k/c}}{4^{6k}}$$

There are at most m^2 pairs of locations and at most n^3 triplets of probes. Thus, the probability is bounded by

$$m^2 n^3 \frac{e^{12k/c}}{4^{6k}} \leq \frac{\alpha^2 \beta k e^{12k/c}}{4^k} << 1$$

for c large enough.

Hence, the probability that throughout the algorithm, there are three fooling probes arising from the same location, is negligible. ■

4.2 Correctness of Algorithm B

Now we have the tools to prove the correctness of algorithm B in the various cases:

Theorem 4.5 *The probability that algorithm B fails is bounded by an arbitrary small constant, if the length of the sequence is $m = O(\frac{4^k}{k})$ and the number of probes is $n = \Theta(k4^k)$, assuming there are no hybridization errors.*

Proof: Since there are no errors, the correct base has always at least k supporting probes. Fix an incorrect base, and let Y be the number of supporting probes of this base. The algorithm fails if at some point $Y \geq k$. Therefore, we would like to show that for some $\epsilon < 1$,

$$Pr[Y \geq k] < \frac{\epsilon}{m}$$

For $k > 3$, if there are k fooling probes they can be partitioned into subsets of sizes 1 or 2 of probes which come from the same location (this is true since by Claim 4.4, the probability of 3 fooling probes that arise from the same location is negligible). By Claim 4.1, the probability of a single fooling probe is bounded by α and by Claim 4.3, the probability of two fooling probes that arise from the same location is bounded by α^2. The number of possibilities for partitions into subsets is the k^{th} Fibonacci number. Hence,

$$Pr[Y \geq k] \leq \alpha^k \cdot \frac{1}{\sqrt{5}} \left(\left(\frac{1+\sqrt{5}}{2} \right)^k - \left(\frac{1-\sqrt{5}}{2} \right)^k \right) \leq \frac{\epsilon}{m}$$

for α small enough.

Hence, for any ϵ, one can get $m = O(\frac{4^k}{k})$ and $n = \Theta(k4^k)$ with the desired failure bound. ■

Note that by increasing k, one can get any derived failure probability even if $m = \alpha \frac{4^k}{k}$ and $n = k4^k$.

Theorem 4.6 *Suppose there are both false negatives and false positives, the length of the sequence is* $m = O((1-q)\frac{4^k}{k})$ *and the number of probes is* $n = \Theta(\frac{k4^k}{1-q})$, *where* q *is the rate of false negatives and the rate of false positives is* $p \leq \frac{1-q}{430}$. *Then, the probability that algorithm B fails is bounded by an arbitrary small constant.*

Proof: Let X be a random variable that counts the number of supporting probes for the correct base. Fix an incorrect base, and let Y be the number of supporting probes of this base. Clearly, $Y = Y_1 + Y_2$, where Y_1 is the number of supporting probes of the false extension that are fooling probes, and Y_2 is the number of supporting probes arising from false positives. The algorithm fails if at some point $Y \geq X$. Therefore, we would like to show that for some $\epsilon < 1$,

$$Pr[X \leq Y_1 + Y_2] < \frac{\epsilon}{m}$$

We first bound X. In this case, X is binomial, as in our model, the errors are independent. Thus, $X \sim B(\frac{5k}{1-q}, 1-q)$ and therefore, by Lemma 2.5,

$$Pr[X < k] = Pr[X - 5k < -4k] < e^{-1.6k} < \frac{\epsilon}{3m}$$

Y_1 is the same random variable as Y in the proof of Theorem 4.5 and therefore, $Pr[Y_1 \geq \frac{k}{2}] < \frac{\epsilon}{3m}$, for α small enough. Y_2 is clearly bounded by $\frac{5k}{1-q}$. As we need to bound the probability that Y_2 is large, we can assume that $Y_2 \sim B(\frac{5k}{1-q}, p)$. The expectation of Y_2 is $\frac{5kp}{1-q} \leq \frac{k}{86}$. By Lemma 2.4,

$$Pr[Y_2 > \frac{k}{2}] = Pr[Y_2 > 43 \cdot \frac{k}{86}] < \left((e/43)^{\frac{1}{2}} e^{-\frac{1}{86}}\right)^k < \frac{\epsilon}{3m}$$

Hence,

$$Pr[X \leq Y] \leq Pr[X < k] + Pr[Y_1 > k/2] + Pr[Y_2 > k/2] < \frac{\epsilon}{m}$$

■

Remark 4.7 The assumption that the false positive rate is very low has the following *biochemical* justification. By increasing the hybridization stringency, the number of false positives can be decreased at the expense of increasing the false negative rate, which has a small effect on the success probability (the length of the reconstructed sequence depends linearly on the rate of false negatives).

5 Experimental Results

We implemented algorithm B and tested it in simulations with noisy data. The implementation is fast: on a regular workstation, reconstruction requires less than a second even with $n = 4^{10}$ probes and targets of length 35,000.

We tested the algorithm on various types of sequences:

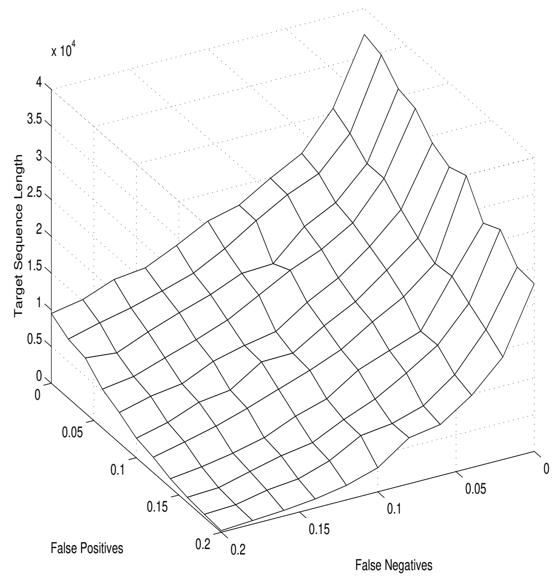
- Randomly generated sequences.
- DNA sequences, which originate from bases 20,000,000 - 40,000,000 of human chromosome 10 [16].
- DNA coding sequences, which were formed by a concatenation of the unigene file Hs.seq.uniq [17].

For some of the sequences we tested also the success of almost perfect reconstruction. A reconstruction is considered almost perfect if up to 0.5% of the bases of the sequence are incorrect.

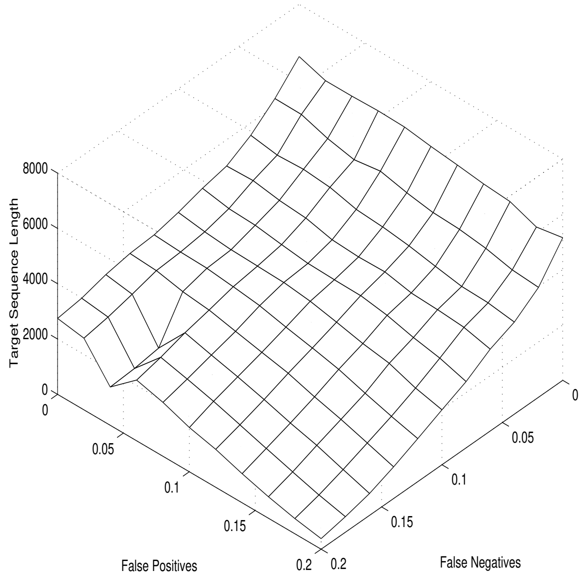
Figure 3 summarizes the performance as a function of the noise parameters. Behavior as a function false positive rate is surprisingly good and extends far beyond the range of the theoretical analysis: In the absence of false negative errors, even if $p = 0.2$, we can reconstruct random sequences of length over $22,000$. Tolerance to false negatives is a bit lower. The simulations indicate a nearly linear correlation between the length of reconstructible sequence and p. As expected, the performance on randomly generated sequences is the best, and coding sequences behave better than arbitrary DNA sequences, since they are much less repetitive. An interesting feature of the algorithm is that it tends to make few and local errors (if any): if we tolerate a very small number of mismatches (almost perfect reconstruction), then, for example, the average length of reconstruction with 0.2 error rates goes up from 379 to 2380.

Figure 4 measures the effect of the parameter c that reflects the length of the probes (and the sparsity of specified positions in them), for various sequences. The increase in c improves the results for all types of sequences. We do not have a complete explanation to the jump observed for random sequences. Our theoretical analysis (section 4.2) requires that c is sufficiently large, and perhaps $c < 3$ suffices, at least for almost perfect reconstruction.

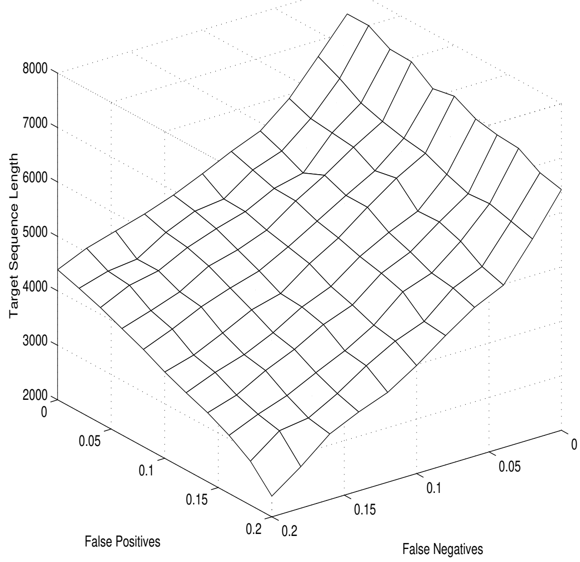
Figure 5 compares the performance of our algorithm to the only other experimental report on performance on error-prone data using universal probes. Doi and Imai [8] developed an extension to the Preparata et al. scheme which handles noisy spectra. For the comparison, we used the same number of probes reported by [8] and plotted their and our experimental success rate as a function of the target length. For the error-less case the algorithm of Preparata et al. is superior: For example, using 4^{10} probes, that algorithm reconstructs sequences of length 70000 bases, while ours can only reconstruct sequences of length 37000. However, the results in Figure 5 show that our algorithm is much more robust to errors. For example, the length reconstructed with 97% success rate improves the results of [8] by a factor of 4. The Doi-Imai algorithm is superior for success rate bellow 50%, but obviously, this is not the range of interest.



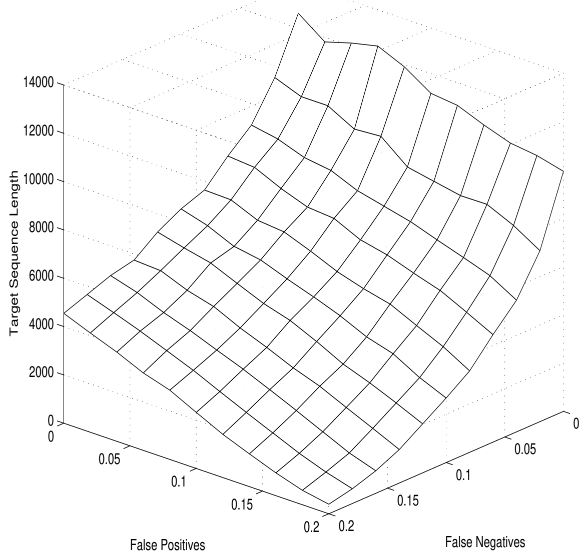
(a) Randomly generated sequences



(b) DNA sequences



(c) DNA sequences (almost perfect reconstruction)



(d) DNA coding sequences

Figure 3: The length of the reconstructed target sequences as a function of the error rates, for various sequences. The z axis measures L, the length of the longest targets correctly reconstructed in at least 90% of the cases. In all cases we used $k = 8$,$n = 4^{10}$ probes and $c = 10$. In (a), we used a sample of 200 random sequences to obtain statistics on each data point. In (b),(c) and (d), we used non-overlapping substrings of length L, covering together the DNA sequence source, to obtain statistics on each data point.

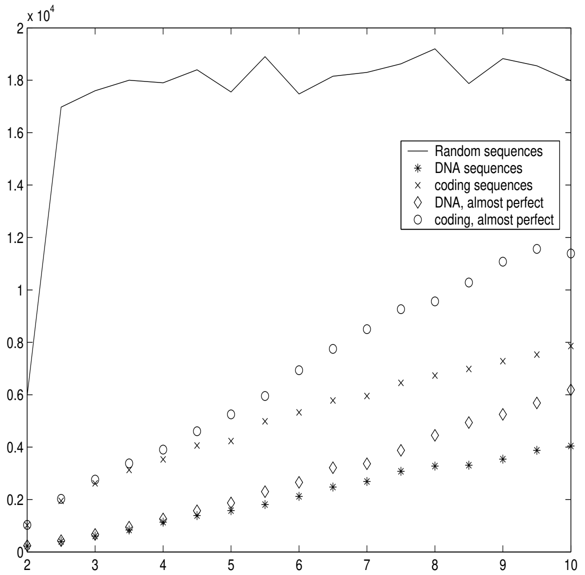


Figure 4: The length of the reconstructed target sequences as a function of the sparsity of specified symbols in the probes (the parameter c), for various sequences. The y axis measures L, the length of the longest targets correctly reconstructed in at least 90% of the cases. A sample of 200 sequences was used to obtain statistics on each data point. In all cases we used $k = 8$,$n = 4^{10}$ probes and $p = q = 0.05$.

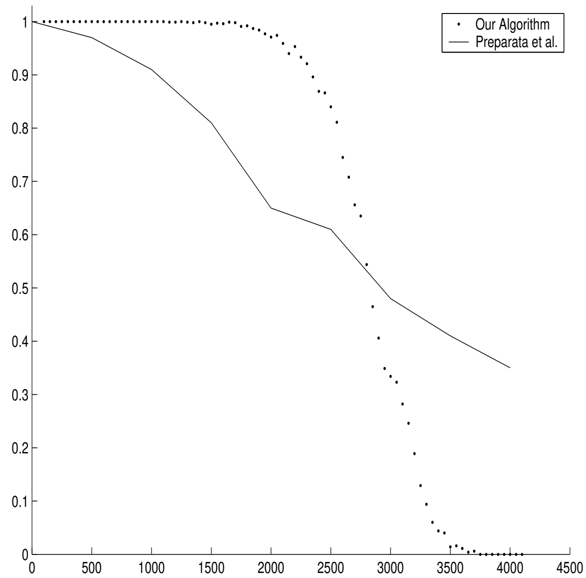


Figure 5: Reconstruction success rate as a function of target length by our algorithm (solid line) and the extension of Doi and Imai [8] to the algorithm of Preparata et al. (dotted line). The success rate is the fraction of correctly reconstructed sequence targets. A sample of 200 random sequences was used to obtain statistics on each data point. In both simulations there were 4^8 probes, the error level was $p = q = 0.001$. The probe sparsity was $c = 4$.

6 Concluding Remarks

In this paper we introduced a new approach to design DNA arrays coping with false positive and negative errors. To the best of our knowledge this is the first time a theoretical analysis was done for such a model. We show that both theoretically, and in practice (using simulations), the sensitivity of our algorithm to errors is smaller, in comparison with previous algorithms.

We intend to continue our work, and to analyze the algorithms assuming other models of hybridization errors. We believe that our results carry over to the model of [14], perhaps after minor algorithmic changes.

A natural extension of algorithm B is an algorithm similar to algorithm A, that is, instead of counting the number of supporting probes for only four possible extensions, we could count the number of supporting probes of a set of possible paths that extend the current reconstructed sequence. It is likely that this algorithm will give better results in practice than algorithm B.

Although universal bases have been generated successfully in the laboratory [15], it is unclear yet if long probes that contain many universal bases hybridize reliably. The fact that low sparsity suffices ($c = 4$ when $k = 6$) is encouraging.

Acknowledgements

Shay Halperin was supported by an Eshkol Postdoctoral Fellowship of the Israel Ministry of Science, Culture and Sport. Ron Shamir was supported in part by a grant from the US-Israel Binational Science Foundation (BSF). We would like to thank Dekel Tsur for comments on the manuscript.

References

[1] N. Alon and J. H. Spencer. *The Probabilistic Method.* John Wiley and Sons, Inc., 2000.

[2] R. Arratia, D. Martin, G. Reinert, and M. S. Waterman. Poisson process approximation for sequence repeats, and sequencing by hybridization. *J. Comp. Biol.*, 3(3):425–463, 1997.

[3] W. Bains and G. C. Smith. A novel method for nucleic acid sequence determination. *J. Theor. Biol.*, 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. Comp. Biol.*, 8(4):361–371, 2001.

[5] J. Blazewicz, P. Formanowicz, F. Glover, M. Kasprzak, and J. Weglarz. An improved tabu search algorithm for DNA sequencing with errors. In *Proceedings of the III Metaheuristics International Conference MIC'99, Angra dos Reis*, pages 69–75, 1999.

[6] J. Blazewicz, P. Formanowicz, K. Kasprzak, W. T. Markeiwicz, and J. Weglarz. DNA sequencing with positive and negative errors. *J. Comp. Biol.*, 6(1):113–123, 1999.

[7] J. Blazewicz, J. Kaczmarek, K. Kasprzak, W. T. Markeiwicz, and J. Weglarz. Sequential and parallel algorithms for DNA sequencing. *CABIOS*, 13:151–158, 1997.

[8] K. Doi and H. Imai. Sequencing by hybridization in the presence of hybridization errors. In *Proceedings of the Workshop on Genome Informatics (GIW '00)*, volume 11, pages 53–62, 2000.

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

[10] M. Dyer, A. Frieze, and S. Suen. The probability of unique solution of sequencing by hybridization. *J. Comp. Biol.*, 1:105–110, 1994.

[11] A. Frieze and B. Halldorsson. Optimal sequencing by hybridization in rounds. *J. Comp. Biol.*, 9(2):355–369, 2002.

[12] E. Halperin, S. Halperin, T. Hartman, and R. Shamir. Handling long targets and errors in sequencing by hybridization. In *Proceedings of the Sixth Annual International Conference on Computational Moleculaer Biology (RECOMB 2002)*, pages 176–185, 2002.

[13] S. Hannenhalli, W. Feldman, H. F. Lewis, S. S. Skiena, and P. A. Pevzner. Positional sequencing by hybridization. *CABIOS*, 12(1):19–24, 1996.

[14] R. J. Lipshutz. Likelihood DNA sequencing by hybridization. *J. Biomol. Str. Dyn.*, 11:637–653, 1993.

[15] D. Loakes and D. M. Brown. 5-nitroindole as a universal base analogue. *Nucl. Acids Res.*, 18:2653–2660, 1990.

[16] National Center for Biotechnology Information. Human genome sequencing, 2002. `http://www.ncbi.nlm.nih.gov/genome/seq/`.

[17] National Center for Biotechnology Information. The unigene database, 2002. `http://www.ncbi.nlm.nih.gov/UniGene/`.

[18] I. Pe'er and R. Shamir. Spectrum alignment: Efficient resequencing by hybridization. In *Proceedings of the 8th International Conference on Intelligent Systems for Molecular (ISMB-00)*, pages 260–268, Menlo Park, CA, August 16–23 2000. AAAI Press.

[19] P. A. Pevzner. l-tuple DNA sequencing: computer analysis. *J. Biomol. Struct. Dyn.*, 7:63–73, 1989.

[20] P. A. Pevzner. *Computational Molecular Biology: An Algorithmic Approach.* MIT Press, 2000.

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

[22] 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. Biomol. Struct. Dyn.*, 9:399–410, 1991.

[23] P. A. Pevzner and M. S. Waterman. Open combinatorial problems in computational molecular biology. In *Proceedings of the Third Israel Symposium on Theory of Computing and Systems (ISTCS)*, pages 158–173, 1995.

[24] V. T. Phan and S. S. Skiena. Dealing with errors in interactive sequencing by hybridization. *Bioinformatics*, 17(10):862–870, 2001.

[25] F. Preparata, A. Frieze, and E. Upfal. Optimal reconstruction of a sequence from its probes. *J. Comp. Biol.*, 6(3-4):361–368, 1999.

[26] F. Preparata and E. Upfal. Sequencing by hybridization at the information theory bound: An optimal algorithm. *J. Comp. Biol.*, 7(3-4):621–630, August 2000.

[27] R. Shamir. Algorithms in molecular biology: Lecture notes, Fall 2001-2002. Available at `http://www.math.tau.ac.il/~rshamir/algmb/01/algmb01.html`.

[28] R. Shamir and D. Tsur. Large scale sequencing by hybridization. *J. Comp. Biol.*, 9(2):413–428, 2002.

[29] S. S. Skiena and G. Sundaram. Reconstructing strings from substrings. *J. Comp. Biol.*, 2:333–353, 1995.

[30] 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.