BIOINFORMATICS

Vol. 17 no. 10 2001
Pages 1–9

Dealing with errors in interactive sequencing by hybridization

Vinhthuy T. Phan and Steven S. Skiena

Department of Computer Science, State University of New York, Stony Brook, NY 11794-4400, USA

Received on April 20, 2001; revised and accepted on July 2, 2001

ABSTRACT

Motivation: A realistic approach to sequencing by hybridization must deal with realistic sequencing errors. The results of such a method can surely be applied to similar sequencing tasks.

Results: We provide the first algorithms for interactive sequencing by hybridization which are robust in the presence of hybridization errors. Under a strong error model allowing both positive and negative hybridization errors without repeated queries, we demonstrate accurate and efficient reconstruction with error rates up to 7%. Under the weaker traditional error model of Shamir and Tsur (*Proceedings of the Fifth International Conference on Computational Molecular Biology (RECOMB-01)*, pp 269–277, 2000), we obtain accurate reconstructions with up to 20% false negative hybridization errors. Finally, we establish theoretical bounds on the performance of the sequential probing algorithm of Skiena and Sundaram (*J. Comput. Biol.*, **2**, 333–353, 1995) under the strong error model.

Availability: Freely available upon request.

Contact: skiena@cs.sunysb.edu

INTRODUCTION

Sequencing By Hybridization (SBH; Bains and Smith, 1988; Dramanac and Crkvenjakov, 1987; Fodor *et al.*, 1991; Lysov *et al.*, 1988; Pevzner and Lipshutz, 1994) is an approach to DNA sequencing quite distinct from the traditional Gilbert–Sanger method. In SBH, a set of single-stranded fragments are attached to a substrate, forming a *sequencing chip*. A solution of single-stranded target DNA fragments are exposed to the chip. These fragments hybridize with complementary fragments on the chip, and the hybridized fragments can be identified using a fluorescent or phosphorescent dye. Each hybridization (or the lack thereof) determines whether the string represented by the fragment is or is not a substring of the target. The target DNA can now be sequenced based on the constraints of which strings are and are not substrings of the target. Two surveys (Chetverin and Kramer, 1994; Pevzner and Lipshutz, 1994) give an excellent overview of SBH, both technologically and algorithmically.

There have recently been several interesting algorithmic breakthroughs in the theory of SBH (Ben-Dor *et al.*, 1999; Frieze and Halldorsson, 2001; Preparata and Upfal, 2000; Shamir and Tsur, 2001). However, it is fair to say that experimental interest in hybridization arrays has shifted almost entirely away from *de novo* sequencing to the analysis of gene expression and identifying single nucleotide polymorphisms in previously sequenced organisms. Much of the anticipated need for SBH has receded in the face of advances in large-scale, automated sequencing machines based on traditional shotgun approaches, culminating in the successful effort to sequence the human genome.

Despite the success of Gilbert–Sanger based sequencing approaches, we believe there remain windows of opportunity where variants of SBH may be an economically viable technology; specifically in (1) re-sequencing variants of previously sequenced organisms, and (2) gap closing efforts in larger projects. Two technical hurdles which have held back the use of SBH for *de novo* sequencing must be overcome:

- *Error rates*—perhaps the most serious problem has been the substantial error rate associated with hybridization probes. Both random and systematic hybridization errors occur frequently in experimental hybridization data. These problems prove less intractable in re-sequencing applications (such as the design of a custom HIV chip) because redundant reference probes can be effectively employed, and probe lengths can be made long enough to increase hybridization reliability.
- *Combinatorial constraints*—the classical sequencing chip $C(m)$, contains all 4^m single-stranded oligonucleotides of length m. For example, in $C(8)$ all $4^8 = 65\,536$ octamers are used. The classical chip $C(8)$ suffices to reconstruct 200 nucleotide long sequences in only 94 of 100 cases (Pevzner *et al.*, 1991), even in error-free experiments. Unfortunately, the length of unambiguously reconstructible sequence

grows slower than the size (i.e. area) of the chip. Thus exponential growth inherently limits the length of the longest reconstructible sequence by traditional SBH. More recent array designs based on gapped probes (Preparata and Upfal, 2000) prove much more efficient, approaching the information theory limit for array designs. However, it is unclear how accurately such gapped probes detect hybridizations in practice, and how robust such chip designs can be in the face of error.

We (Margaritis and Skiena, 1995; Skiena, 1997) have proposed a different approach to SBH which permits the sequencing of arbitrarily large fragments without the inherently exponential chip area of SBH, while retaining the massive parallelism which is the primary advantage of the technique. Our approach uses *interaction* to reduce the required amount of work. After beginning with an experiment using a prefabricated sequencing chip (such as $C(8)$), we take the results from this experiment and use them to design a customized sequencing chip to resolve some of the remaining ambiguities. We repeat this process until the sequence is determined. The key issue is demonstrating that we can design arrays completing the job using as few rounds of queries as possible. Margaritis and Skiena (1995) use less than ten rounds to reduce the total number of queries by several orders of magnitudes on reasonable fragment sizes, under the assumption of error-less queries. The potential savings increase very rapidly with the length of the fragment to be sequenced.

The critical issue in establishing the practicality of Interactive Sequencing By Hybridization (ISBH) has been reducing the cost of custom-designed sequencing chips. Fortunately, there has been significant progress in this direction in recent years. Machines for producing oligonucleotide arrays using ink-jet printer technology have been pioneered by Blanchard *et al.* (1996) of Rosetta Inpharmatics. Agilent Technologies is now manufacturing ink-jet array machines on a commercial scale. Hood (2000) claims the technology to build arrays with 100 000 probes, each of length up to 40–50 bases in a 2 h cycle time for about US $80 an array on a US $50 000 synthesizer. Other proposed technologies, such as LCD photo-lithography masks for Affymetrics-type arrays (Fodor *et al.*, 1991) and pixel-addressable Southern Array Makers (Bradley and Skiena, 1997; Wehnert *et al.*, 1994), also show promise.

For this reason, we believe now is the time to revisit ISBH to establish whether the technology is potentially viable under realistic error models. We also believe that this work is helpful in demonstrating the impact that hybridization errors will have on *any de novo* SBH procedure, and thus focusing the attention of the algorithmic SBH community towards designing and analyzing approaches more robust in the face of hybridization errors.

The outline of this paper is as follows. In the next section, we present an overview of the state of the art in traditional DNA SBH and ISBH. We then introduce an error model for ISBH simulations and propose a reconstruction algorithm robust in the face of such errors. Finally, we report on the performance of our reconstruction algorithm in simulations over a range of error rates. Our results are encouraging. As shown in Table 1 our average number of probes and rounds, with error rates of up to 7%, are only modestly larger than the error-less case of these sequences.

SEQUENCING BY HYBRIDIZATION, WITH INTERACTION

SBH is an alternate approach to DNA sequencing which offers the potential of reduced cost and higher throughput over traditional gel-based approaches. Strezoska *et al.* (1991) accurately sequenced 100 base pairs of a known sequence using hybridization techniques, although the approach was proposed independently by several groups, (Bains and Smith, 1988; Dramanac and Crkvenjakov, 1987; Lysov *et al.*, 1988; Macevicz, 1989; Southern, 1988). Crkvenjakov's and Drmanac's laboratories report sequencing a 340 base-pair fragment in a blind experiment (Pevzner and Lipshutz, 1994). More recently, Morris and Huang (1999) performed *de novo* sequencing experiments using prefabricated hybridization arrays of all 10 mers, proving able to sequence short (125 bp) fragments of mitochondrial DNA in the face of hybridization errors.

Interpreting the data is where the algorithmic aspect of SBH arises. The outcome of an experiment with a classical sequencing chip $C(m)$ is a probability that each of the 4^m strings is a substring of the underlying sequence S. In an experiment without error, the probabilities will all be 0 or 1, so each m-nucleotide fragment of S is unambiguously identified.

Efficient algorithms exist for finding the shortest string consistent with the results of a classical sequencing chip experiment. In particular, Pevzner's algorithm for sequencing chip reconstruction (Pevzner, 1989) is based on finding Eulerian paths in a subgraph of the de Bruijn digraph (de Bruijn, 1946). Pevzner also gives an algorithm based on maximum flow to reconstruct sequences from spectrum which can contain false negatives, but not false positives. Blazewicz *et al.* (1999) present an algorithm for handling both positive and negative errors based on the selective traveling salesman problem, and demonstrate the ability to give accurate reconstructions of up to 400 bp sequences using 10 mer probes with up to 10% false positive and negative *hybridizations*. However, this model grossly undercounts the number of actual false positives if each of the 4^{10} 10 mer probes has a 10% false positive rate. Indeed, false positives can so overwhelm SBH algorithms that Shamir and Tsur (2001) only consider false negative errors in their work.

Table 1. Performance of the adaptive algorithm on GenBank sequences. Repeats counts the number of repeated 30 mers in the target

Sequence	Genome		Errorless case		7% error rate		
	Length	Repeats	Rounds	Probes	Rounds	Phase 1	Phase 2
Human alpha globin	12 847	2238	12	125 546	16	311 400	491 200
Human beta globin	18 060	0	11	167 722	22	390 500	524 000
Chicken collagen	21 180	0	9	153 836	13	487 400	389 500
Human HIV virus	9 718	407	11	83 954	12	247 700	167 500
Bacteriophage lambda	48 502	0	11	386 218	12	1520 000	720 000
Mouse mitochondrion	16 295	0	10	120 030	11	591 900	376 500
Rat myosin heavy chain	25 759	45	11	235 652	18	576 400	674 000
Rabies virus	11 928	0	11	99 167	11	365 400	163 600
Human rhinovirus type 14	7 212	0	9	52 634	13	164 500	86 600
Simian Virus 40	5 243	41	11	48 003	12	117 800	92 300
Drosophila white locus	14 245	0	10	113 202	19	486 300	389 800

Every hybridization determines, for a given string s, whether s a substring of an unknown string S. We are not told where s occurs in S, nor how many times it occurs, just whether or not s is a substring of S. To understand the algorithmics of ISBH, we must first understand the complexity of reconstructing strings from such substring queries.

Skiena and Sundaram (1995) studied the complexity of *sequentially* reconstructing unknown strings from substring queries. Specifically, we showed that $(\alpha - 1)n + \Theta(\alpha\sqrt{n})$ queries are sufficient to reconstruct an unknown string, where α is the alphabet size and n the length of the string, matching the information-theoretic lower bound for binary strings. Further, we show that $\sim\alpha n/4$ queries are necessary, which is within a factor of 4 of the upper bound for larger alphabets. However, achieving a high degree of parallelism is critical for this approach to lead to a practical method of DNA sequencing.

Margaritis and Skiena (1995) studied the complexity of reconstructing unknown strings from rounds of parallel substring queries. We showed a wide range of trade-offs between the number of rounds of substring queries and the number of queries per round sufficient to determine an unknown string of length n on an alphabet of size α. These ideas were further developed by Kruglyak (1998). Further, Margaritis and Skiena (1995) give a strategy which uses (with probability $1-1/n^{\epsilon}$) $O(\alpha\epsilon \lg n)$ rounds of n queries per round. This has recently been improved to a constant number of rounds under a more powerful probing model (Frieze and Halldorsson, 2001).

In Table 1, we report on the number of rounds required to determine actual DNA sequences from the original ISBH paper of Margaritis and Skiena (1995). In this paper, we will analyze these sequences under substantial error conditions.

RECONSTRUCTION IN THE PRESENCE OF ERRORS: AN ERROR MODEL

In this section, we begin by discussing our proposed error model before explaining the details of the two phases of our reconstruction algorithm.

There are two possible types of hybridization errors. A probe s is a *false positive* if an experiment reports the presence of s even though s is not actually a substring of the target DNA sequence S. A probe s is a *false negative* if an experiment reports the absence of s even though s is a substring of S.

Establishing the right error model for hybridization is a subtle problem. Certain errors are *random*, meaning that they would disappear when repeating the experiment. Sources of random error include contamination and suboptimal hybridization conditions. However, many hybridization errors are *systematic*, meaning that they are likely to repeat each time an experiment is run. For example, palindromic subsequences tend to form secondary structures which interfere with hybridization. In previously published simulations of SBH with error (Blazewicz *et al.*, 1999), no systematic error was considered.

The reliability of a probe depends upon its binding energy to the target, and is a function of (1) the length of the probe, (2) the oligonucleotide content of the probe, and (3) the distribution of similar sequences in the target. In general, longer (length 20) probes are more reliable than shorter ones (length 8), although C–G bonds are roughly twice as strong as A–T bonds. False positives tend to occur when there are minor (say 1-base) mismatches between probe and target, particularly when the mismatch occurs at the end of the probe. Accurately capturing these subtleties requires a sophisticated error model reflecting detailed understanding of local laboratory procedures and conditions.

To capture some of the effect of systematic errors

and to avoid the dependence of errors of subsequent queries on the same probe, we forbid our strategy from ever asking a given probe more than once. Further, we forbid our strategy from asking probes of different lengths in a single round. The optimal hybridization and stringency conditions vary considerably by probe lengths, and heterogeneous-length probes cannot be expected to perform accurately.

To capture the effect of random errors, we parameterize the experimental conditions by positive and negative hybridization accuracy probabilities P_+ and P_-, respectively. Each probe or query is a random variable $[s \in S?]$ for a string s, which assumes values Y or N according to the following probability distribution:

$$[s \in S?] = \begin{cases} Y & \text{with probability } P_+, \text{ if } s \in S \\ N & \text{with probability } 1 - P_+, \text{ if } s \in S \\ Y & \text{with probability } 1 - P_-, \text{ if } s \notin S \\ N & \text{with probability } P_-, \text{ if } s \notin S. \end{cases}$$

It is useful to think of a round of hybridization as asking an oracle a collection of questions (probes), to which he answers yes (positive hybridization signal) or no (negative signal) subject to a given error rate.

A SEQUENTIAL PROBE STRATEGY

We first look at the case where we are allowed to ask a single probe in a round, seeking to reconstruct the sequence using the smallest number of probes. Skiena and Sundaram (1995) provide an optimal algorithm for the errorless case, which we now extend and analyze under our error model. Suppose that we start with an initial string $S = s[1 \cdots M']$ which we know with high probability to be a substring of the DNA sequence (such an S can be found by using $O(n)$ probes of length $\log_4 n$, for example). We propose a simple algorithm that extends S one character at a time until the sequence is complete.

Algorithm 1 Extend($S = s[1 \cdots M']$)

for all $C \in \Sigma$ **do**
  Determine threshold V;
  Probes=$\bigcup_{1 \le i \le M < M'} \{ s[i \cdots M']C \}$;
  Ask all M probes;
  if (the number of yes's $\ge V$) **then**
    return $S[1 \cdots M']C$;
  end if
end for
STOP;

Note that a false positive error occurs when we extend an incorrect character. A false negative error occurs when we miss extending a correct character. We assume that the false positive and false negative errors occur with the same probability P ($P = 1 - P_+ = 1 - P_-$). We will show that with the threshold of acceptance V at precisely half the number of probes attempted, the probability of making a positive or negative error is small. We assume the probe length (M') is long enough to ensure that repetitions of this length in the target sequence are unlikely. Thus positive probes can only be accounted for by one position in the target sequence.

LEMMA 1. *If the lie probability $P < \frac{1}{2}$ and $V = \frac{M}{2}$, then the extension algorithm fails with probability less than $(4P(1-P))^{\frac{M}{2}}$.*

PROOF. Let Y and N be the random variable indicating the number of yes's and no's in M trials. By our assumption, either the probes are M substrings (M_Y) or M non-substrings ($\neg M_Y$). We have:

- $\Pr(\text{false positive}) = \Pr(Y \geqslant V | \neg M_Y)$.
- $\Pr(\text{false negative}) = \Pr(Y < \frac{M}{2} | M_Y) = \Pr(N \geqslant M - \frac{M}{2} | M_Y)$.

Since in both cases, the lie probability is the same, P, both error probabilities are equal to $\Pr(X \geqslant \frac{M}{2})$, where X indicates the number of lies in M trials, and $\Pr(X_i) = P$.

Since e^x is strictly increasing, we have $\Pr(X \geqslant \frac{M}{2}) = \Pr(e^{\delta X} \geqslant e^{\delta \frac{M}{2}})$ for all $\delta > 0$. By Markov's inequality, we have $\Pr(e^{\delta X} \geqslant e^{\delta \frac{M}{2}}) \leqslant e^{-\delta \frac{M}{2}} \cdot E[e^{\delta X}]$. Further,

$$E[e^{\delta X}] = E[e^{\sum_{i=1}^{M} \delta X_i}] = E\left[\prod_{i=1}^{M} e^{\delta X_i}\right] \quad (1)$$

$$= \prod_{i=1}^{M} E[e^{\delta X_i}] = (Pe^{\delta} + (1-P))^M \quad (2)$$

assuming the X_is are mutually independent. Therefore, $\Pr(X \geqslant \frac{M}{2}) \leqslant e^{-\delta \frac{M}{2}} \cdot (Pe^{\delta} + (1-P))^M$, for all $\delta > 0$. The right-hand side achieves its minimum at $\delta = \ln \frac{1-P}{P} > 0$. Using this value, we have $\Pr(\text{false positive}) = \Pr(\text{false negative}) \leqslant (4P(1-P))^{\frac{M}{2}}$. Finally, $\Pr(\text{error}) = \frac{1}{\alpha} \cdot \Pr(\text{false negative}) + \frac{\alpha - 1}{\alpha} \cdot \Pr(\text{false positive}) \leqslant (4P(1-P))^{\frac{M}{2}}$. □

When $M = (\log_{4P(1-P)} \alpha) \cdot (-4 \log_\alpha n)$, the probability of error of making n extensions (n the length of the DNA sequence) is less than $n \cdot \frac{1}{n^2} = \frac{1}{n}$. To warranty that the M probes are either all substrings or all non-substrings of the DNA sequence, we must start the extension algorithm with a positive substring of length about $M + (\log_\alpha n)^2$.

We show the number of probes needed to extend one character with different levels of accuracy up to 16% of error probability. Generally, the number of probes needed increases exponentially with increasing lie probability.

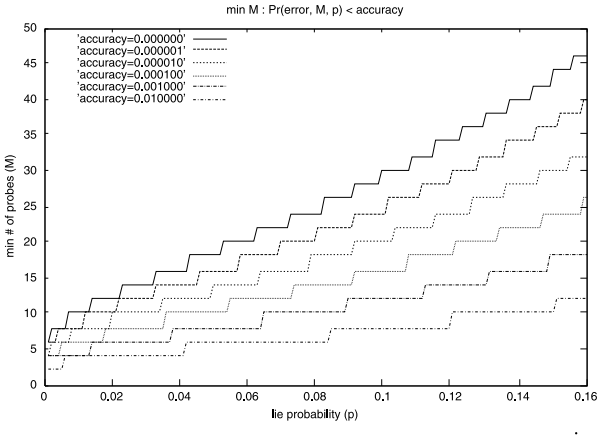

Fig. 1. The number of probes needed to extend 1 symbol as a function of P, at different levels of accuracy.

A PARALLEL PROBE STRATEGY

Determining oligonucleotide content

A typical, error-free SBH algorithm starts by asking $4^{k_{\text{start}}}$ probes of length $k_{\text{start}} = O(\log_4 |S|)$, which results in about $(1-P_-)\cdot(4^{k_{\text{start}}} - |S|)$ false positives; the number of 'false' yes's, therefore, can be really large in comparison to the number of 'true' yes's.

The interactive reconstruction algorithms we explore mirror the basic structure of algorithms developed for the problem without errors (Frieze and Halldorsson, 2001; Margaritis and Skiena, 1995). They consist of two distinct phases. The first phase involves asking relatively short probes and iteratively reconstruct longer strings, to provide the basic oligo content of the target sequence. The second phase attempts to recover missing strings and eliminate the last few false positives, by assembling determined oligos into a subgraph of the *de Bruijn graph*, and proposing longer probes to resolve branch points, joining components, and confirming paths in the graph,

Let $P(v \in S)$ be the probability that v is a substring of the target sequence. In the first phase, we probe all sequences v whose $P(v \in S)$ is above a dynamically changing threshold, and use this new information to update $P(v \in S)$. The start length k_{start} must be large enough that a substantial number of oligos are not present, yet small enough that generating all $4^{k_{\text{start}}}$ probes is feasible. Hence, $k_{\text{start}} = \log|S| + c$ is a good starting length.

When a probe answer to a string v is yes, using Bayes theorem and our currently estimate $P(v \in S) = p$, the updated $P(v \in S)$ is the maximum of p_- and the estimated $P(v \in S|Y)$ which is:

$$\frac{p \cdot P(Y|v \in S)}{p \cdot P(Y|v \in S) + (1-p) \cdot P(Y|v \notin S)}$$

$$= \frac{p \cdot p_+}{p \cdot p_+ + (1-p)\cdot(1-p_-)}.$$

We use a similar rule when the probe answer is no. Further, we use our current knowledge about $P(v \in S)$ to set the basis for querying probes of longer lengths. We assign

$$P(u \in S) = P(P_u \in S) \cdot P(S_u \in S)$$

where length of u is $l+1$, and P_u and S_u are the l-prefix and l-suffix of u. This lets us build on previous results and reduce our queries to a small set of probes. These ideas make up the first phase of the algorithm.

Algorithm 2 First Phase

for all strings v of length k_{start} **do**
  $P(v \in S) = \frac{|S|}{4^{k_{start}}}$;
end for
for $l = k_{start}$ to k_{switch} **do**
  Probe all v of length l such that $P(v \in S) > t$.
  Based on probes' results:
  (**1**) Update $P(v \in S)$ for all v's
  (**2**) Update threshold t
  (**3**) Set $P(v \in S)$ for all v's of length $l+1$.
  (**4**) if($p_- = 1$) Merge all v's whose probe answers are yes's.
end for

Algorithm 3 Second Phase

Let G be the de Bruijn graph of the strings from phase 1, and let LG be its line graph.
repeat
  (**1**) Probe all vertices (i.e. strings) in LG and those necessary to recover false negatives.
  (**2**) Eliminate false positives by looking at:
    (i) vertices whose in/out degree is > 1.
    (ii) all paths in LG of a fixed length.
  (**3**) Recover false negatives by merging all gaps;
  (**4**) Undo merges that might have mistakenly inserted false positives;
until LG still have false positives

Querying paths in the de Bruijn graph

The second phase seeks to remove the last and difficult false positives, as well as close gaps left by false negatives. These two tasks require complementary strategies. We keep the positive sequences from phase 1 as vertices in an explicit de Bruijn graph.

Removal of false positives. To remove false positives, we look at the probe results along simple paths. We would

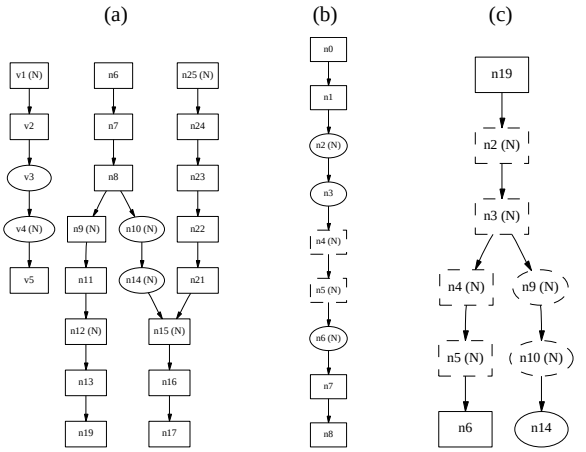

Fig. 2. Some issues dealt with by the algorithm: (a) removal of false positives, (b) over-aggressive merges, and (c) closing gaps by eliminating false positives. The figures illustrate portions of a de Bruijn graph of 30 mers. Probes labeled (N) received No answers, while unlabeled received Yes answers. Oval shapes denote actual false positives, rectangular actual substrings, and dashed nodes false negatives.

anticipate seeing a high frequency of negative answers along any simple path containing a false-positive oligonucleotide, and hence can delete any vertex responsible for more than a given number of negative answers.

A more aggressive removal of false positives depends on our analysis of the structure of the de Bruijn graph. For example, vertices near a branch, i.e. whose in-degree or out-degree is greater than 1, are good candidates of being false positives. To distinguish these false positives from the vertices which are caused by longer repeats in the target sequence, we again look at the frequency of negative answers.

In Figure 2a, for example, we seek to delete false positives such as $n10$ and $n14$—for being near a branch—more aggressively than other false positives $v3$ and $v4$.

Retrieval of false negatives. In this phase, we merge *gaps* in the de Bruijn graph. To merge a gap between two vertices, v_1, v_2, we look at where they overlap and probe all missing strings between them. If enough probes answer affirmatively, we insert these vertices into our de Bruijn graph.

A naive implementation would merge all vertices of out-degree 0 and all vertices of degree 0. However, because of the existence of false positives, an end of a 'real' gap may not have out/in-degree of 0. Therefore, it is important to look not only at the ends, but also to explore further into each contig. parameter d. In Figure 2b, we wish to recover missing strings $n4$, $n5$, this depth parameter allows us to consider merging not only $n3$ and $n6$, but also $n1$ and $n7$.

Other heuristics are necessary to overcome special cases and efficiently trade off probes for accuracy:

- *Limiting fan out*—for any candidate v, there are likely to be few real pairs that v, indeed only one pair if there are no duplicates. Therefore, we consider only a few of the most promising strings to be paired with v.
- *Gap size*—in early iterations, many pairs consist of false positives in one or both ends of a pair. The overlap size of each pair, and hence the gap size, should be treated differently from the same gap size in later iterations. This helps to bound the number of queries as well as to reduce mistaken pairings and hence insertions.
- *Non-optimal overlaps*—a pair u, v can have more than one overlapping alignment. An overlap of length 7 may not give the correct alignment, while in reality an overlap of length 6 does. For this reason, for each pair we look at all overlaps of at least *minimum overlap* in length.
- *Minimum gap*—when a gap is too short, the number of possible confirmation probes which query missing strings may be too small to yield a statistically significant conclusion. Thus the decision to merge for short gaps depends not only on the probe results of the missing strings but also on those of the neighbors.

Because of hybridization errors, we are bound to make mistakes in merging pairs. A mistaken gap insertion may well cause a branch in the de Bruijn graph if a related pair is correctly merged. For example, in Figure 2c, the pair ($n19$, $n14$) is mistakenly merged, inserting false positives $n9$ and $n10$. However, it causes a branch when the pair ($n19$, $n6$) is correctly merged. We *un-merge* suspicious strings by re-examining the branches of the de Bruijn graph after all insertions are complete.

SIMULATION RESULTS

We have simulated our algorithms on the sequences of Table 1 with false negative error probabilities ranging from 1 to 25% for the error model of Shamir and Tsur (2001), as well as the more challenging case of symmetric errors from 2 to 12%. In Figures 3 and 4, each data point shows averaged results of ten different runs, each with different random error. We seek to determine the exact set of 30 mers of the sequence. Due to repeats, this does not necessarily unambiguously determine the DNA sequence, but it is the best achievable given a limit on probe size.

The left chart reflects the cost in terms of the number of probes used, as a function of the simulated error rate. For most instances, more probes are spent in the first phase, which is desirable because these can be best

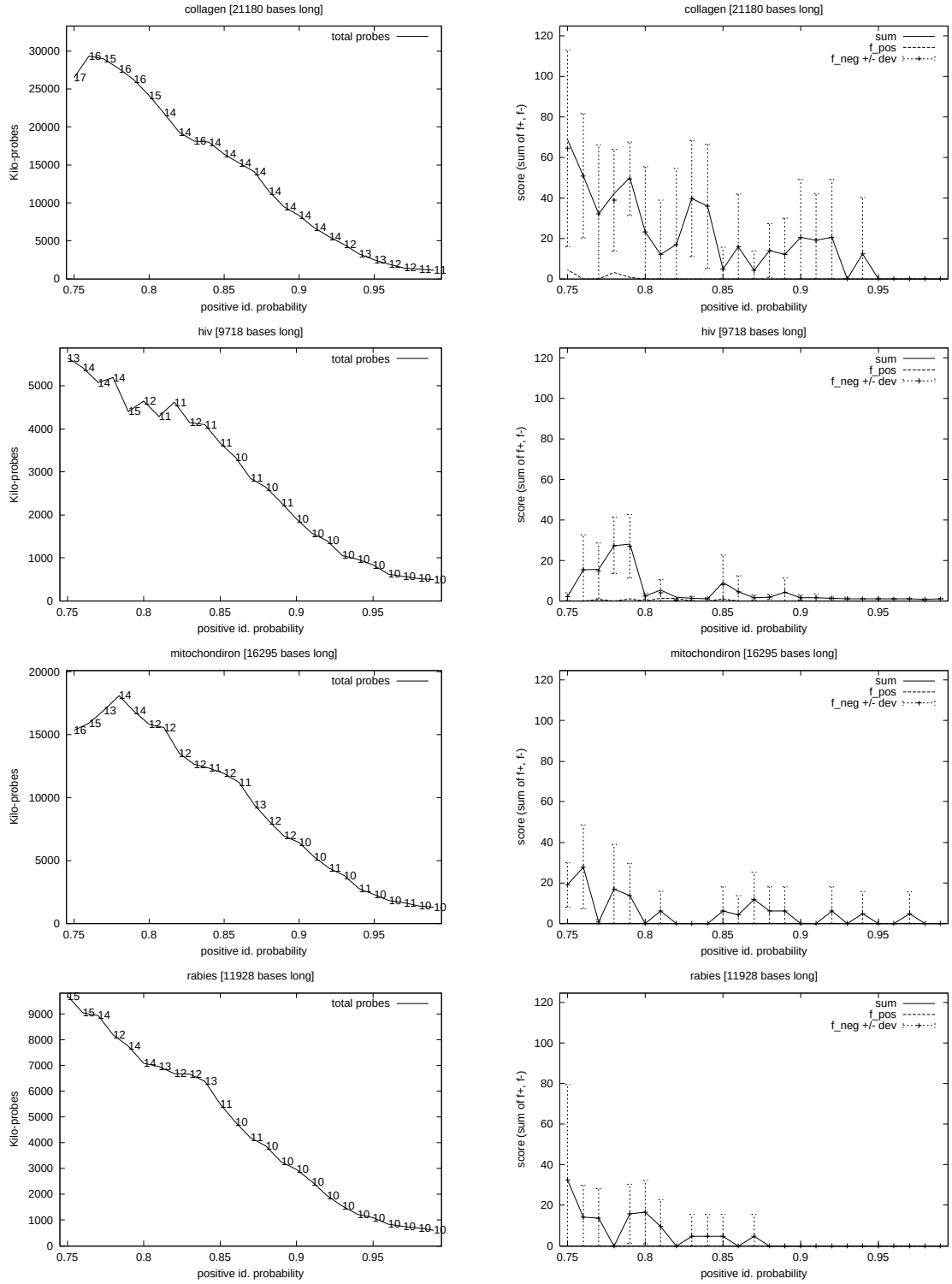

Fig. 3. False positive rate is 0 ($p_- = 1$, p_+ varies). Probe counts (left) and accuracy scores (right) as a function of error. Accuracy given as the sum of false positives and negatives left at the end. The latter is more dominant, so $\pm\sigma$ is also included. Numbers of rounds are labeled at each point. Tested sequences: collagen, HIV, mitochondrion, and rabies.

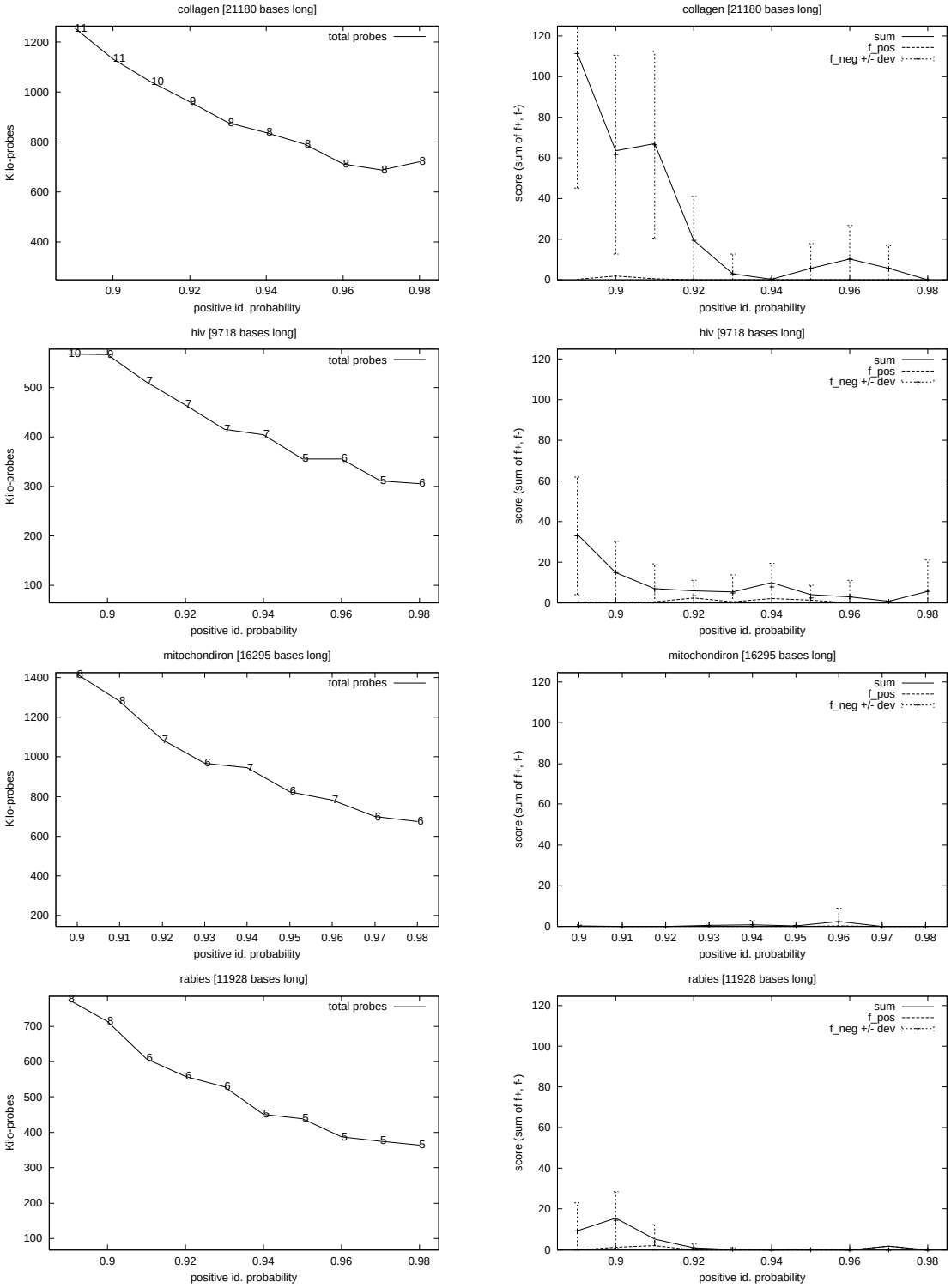

Fig. 4. $p_+ = p_-$. Probe counts (left) and accuracy scores (right) as a function of error. Accuracy given as the sum of false positives and negatives left at the end. The latter is more dominant, so $\pm\sigma$ is also included. Numbers of rounds are labeled at each point. Tested sequences: collagen, HIV, mitochondrion, and rabies.

accommodated by mass-produced arrays of all k mers for small $8 \leqslant k \leqslant 12$. The number of probes in phase 2 fluctuates more, reflecting the fact that certain instances develop larger gaps which are harder to bridge.

The right chart denotes the error score, defined as the sum of false negatives and positives. The number of false positives is very small in almost every case at every error probability, to the extent that it is clearly not a problem. The few remaining false positives tend to be positioned at the end of the DNA sequence, where fewer validation probes exist to remove them. The number of remaining false negatives fluctuates more than that of false positives.

False negatives typically result from gaps in the sequence. Early false negative probes conspire to create gaps too long to accurately bridge. This problem generally becomes acute only when the error rate goes beyond 7% or so. We believe that this bound can be improved somewhat by doing a better job of determining the right value of the parameters for each sequence. In general, sequences with substantial repeats (such as alphaglobin) prove to be somewhat more difficult than others. Each repeat causes a potential confusion in the de Bruijn graph; each missing repeat accounts for as many *holes* in the finished sequence as the number of times it occurs. Our system seemed to do best for mitochondrial DNA.

REFERENCES

Bains,W. and Smith,G. (1988) A novel method for nucleic acid sequence determination. *J. Theor. Biol.*, **135**, 303–307.

Ben-Dor,A., Pe'er,I., Shamir,R. and Sharan,R. (1999) On the complexity of positional sequencing by hybridization. *Lecture Notes in Computer Science* 1645, 88–98.

Blanchard,A., Kaiser,R. and Hood,L. (1996) High-density oligonucleotide arrays. *Biosensors & Bioelectronics*, **11**, 687–690.

Blazewicz,J., Formanowicz,P., Kasprzak,M., Markiewicz,W. and Weglarz,J. (1999) DNA sequencing with positive and negative errors. *J. Comput. Biol.*, **6**, 113–123.

Bradley,J.R. and Skiena,S.S. (1997) Fabricating arrays of strings. In *Proceedings of the First Conference on Computational Molecular Biology (RECOMB 97)*. pp. 57–66.

Chetverin,A. and Kramer,F. (1994) Oligonucleotide arrays: new concepts and possibilities. *Bio/Technology*, **12**, 1093–1099.

de Bruijn,N. (1946) A combinatorial problem. *Proc. Kon. Ned. Akad. Wetensch*, **49**, 758–764.

Dramanac,R. and Crkvenjakov,R. (1987) DNA sequencing by hybridization. Yugoslav Patent Application 570.

Fodor,S., Read,J., Pirrung,M., Stryer,L., Lu,A. and Solas,D. (1991) Light-directed, spatially addressable parallel chemical synthesis. *Science*, **251**, 767–773.

Frieze,A. and Halldorsson,B. (2001) Optimal sequencing by hybridization in rounds. In *Proceedings of the Fifth Conference on Computational Molecular Biology (RECOMB-01)*. pp. 141–148.

Hood,L. (2000) Personal communication.

Kruglyak,S. (1998) Multistage sequencing by hybridization. *J. Comput. Biol.*, **5**, 165–171.

Lysov,Y., Florentiev,V., Khorlin,A., Khrapko,K., Shik,V. and Mirzabekov,A. (1988) Determination of the nucleotide sequence of DNA using hybridization to oligonucleotides. *Dokl. Acad. Sci. USSR*, **303**, 1508–1511.

Macevicz,S. (1989) International Patent Application PS US89 04741.

Margaritis,D. and Skiena,S. (1995) Reconstructing strings from substrings in rounds. In *Proceedings of the Thirty Sixth IEEE Symposium on Foundations of Computer Science (FOCS)*. pp. 613–620.

Morris,M. and Huang,X. (1999) Sequencing by hybridization: theory and practice. *RECOMB Poster Session*.

Pevzner,P. (1989) l-tuple DNA sequencing: computer analysis. *J. Biomol. Struct. Dyn.*, **7**, 63–73.

Pevzner,P., Lysov,Y., Khrapko,K., Belyavski,A., Florentiev,V. and Mizabelkov,A. (1991) Improved chips for sequencing by hybridization. *J. Biomol. Struct. Dyn.*, **9**, 399–410.

Pevzner,P.A. and Lipshutz,R.J. (1994) Towards DNA sequencing chips. In *Proceeding of the Ninteenth International Conference on Mathematical Foundations of Computer Science* 841. pp. 143–158.

Preparata,F.P. and Upfal,E. (2000) Sequencing-by-hybridization at the information-theory bound: an optimal algorithm. In *Proceedings of the Fourth Conference on Computational Molecular Biology (RECOMB-00)*. pp. 245–253.

Shamir,R. and Tsur,D. (2001) Large scale sequencing by hybridization. In *Proceedings of the Fifth International Conference on Computational Molecular Biology (RECOMB-01)*. pp. 269–277.

Skiena,S. (1997) A method of identifying sequence in a nucleic acid target using interactive sequencing by hybridization. US Patent 5 683 881.

Skiena,S. and Sundaram,G. (1995) Reconstructing strings from substrings. *J. Comput. Biol.*, **2**, 333–353.

Southern,E. (1988) United Kingdom Patent Application GB8810400.

Strezoska,Z., Paunesku,T., Radosavljevic,D., Labat,I., Drmanac,R. and Crkvenjakov,R. (1991) DNA sequencing by hybridization: 100 bases read by a non-gel-based method. *Proc. Natl Acad. Sci. USA*, **88**, 10 089–10 093.

Wehnert,M., Matson,R., Rampal,J., Coassin,P. and Caskey,C. (1994) A rapid scanning strip for tri- and di-nucleotide short tandem repeats. *Nucleic Acids Res.*, **22**, 1701–1704.

To be balanced at final stage