A Partial Digest Approach to Restriction Site Mapping

Steven S. Skiena*
Dept. of Computer Science
State University of New York
Stony Brook, NY 11794

Gopalakrishnan Sundaram
Dept. of Applied Mathematics
State University of New York
Stony Brook, NY 11794

September 22, 1993

Abstract

We present a new, practical algorithm to resolve the experimental data in restriction site analysis, which is a common technique for mapping DNA. Specifically, we assert that multiple digestions with a single restriction enzyme can provide sufficient information to identify the positions of the restriction sites with high probability. The motivation for the new approach comes from combinatorial results on the number of mutually homeometric sets in one dimension, where two sets of n points are homeometric if the multiset of $n(n-1)/2$ distances they determine are the same.

Since experimental data contains error, we propose algorithms for reconstructing sets from noisy interpoint distances, including the possibility of missing fragments. We analyze the performance of these algorithms under a reasonable probability distribution, establishing a relative error limit of $r = \Theta(1/n^2)$ beyond which our technique becomes infeasible. Through simulations, we establish that our technique is robust enough to reconstruct data with relative errors of up to 7.0% in the measured fragment lengths for typical problems, which appears sufficient for certain biological applications.

*This work was partially supported by NSF Research Initiation Award CCR-9109289 and by New York Science and Technology Foundation grant RDG-90171.

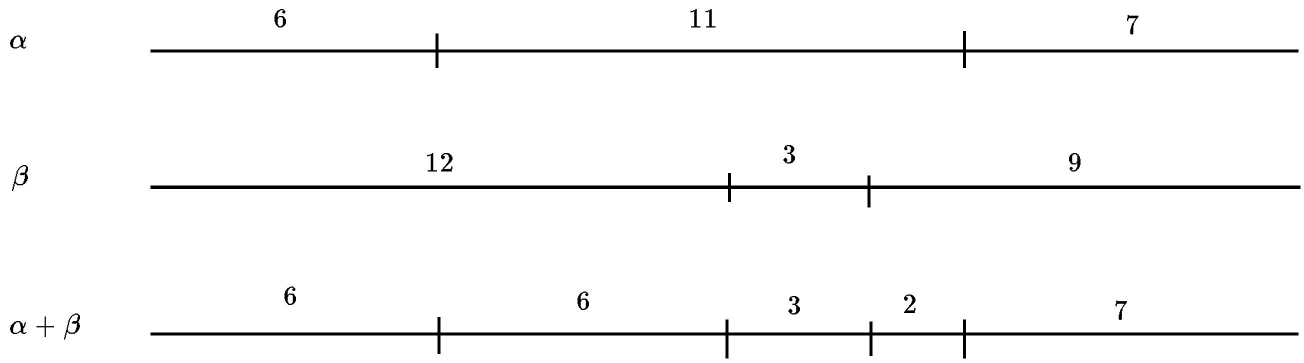

Figure 1: The unique solution to a double digest problem.

1 Introduction

The genetic code of an organism defines a string on the alphabet $\{A, C, G, T\}$, representing the amino acids adenine, cytosine, guanine, and thymine. Restriction enzymes are chemicals which recognize and cut DNA molecules at particular patterns. For example, the enzyme *Eco*RI cuts at the pattern $GAATTC$. Exposing a DNA molecule to a particular restriction enzyme may cut it in one or more places, and measuring the length of the fragments provides information about where the restriction sites are located on the molecule. Reconstructing this information is called *restriction site analysis* and is a common laboratory technique.

Reconstruction can be difficult whenever an enzyme cuts the molecule in more than one place, which is usually the case for sufficiently long molecules. For example, the plasmid pBR322 has been completely sequenced, and [12] provides a list of all restriction sites for 81 different enzymes. The average enzyme cuts in 6.28 sites, with one enzyme cutting in no fewer than 42 separate places.

If we expose a linear molecule to an enzyme and wait for the reaction to complete, each of the n restriction sites are cut and so we obtain $n + 1$ fragments. Such an experiment is known as a *full* or *complete digest*. Any permutation of the fragments resulting from a complete digest is a possible interpretation of the molecule, and so further data is necessary to position the sites. To get additional information, typically an additional enzyme is used. For example, suppose we use enzymes α and β on clones of a linear molecule. Say α cuts the strand into three pieces of length 6, 7, and 11, while β cuts it into three fragments of length 3, 9, and 12. Finally, exposing it to both enzymes yields 2, 3, 6, 6, and 7 unit strands. The only way to reconcile this data is to place the two β sites 3 units apart and between the two α sites, as shown in Figure 1. Thus the restriction sites can be mapped using the distance information. Unfortunately, there is no way to guarantee that some constant number of enzymes k will be sufficient for unique reconstruction, since any unbroken run of three or more sites for a particular enzyme cannot be resolved.

If our sample contains a large number of identical linear molecules, as is obtainable by cloning, and we reduce the time of exposure so that not every site is cut, we would obtain instances of each of the $\binom{n}{2}$ fragments, correspond to fragments between every pair of sites. Such an experiment is called a *partial digest.* Finding the location of sites given the interpoint distances between pairs of sites is known as the turnpike reconstruction problem, and the combinatorial results of Skiena, Smith, and Lemke [18] presented in Section 2 suggest that only carefully constructed examples permit multiple reconstructions consistent with the results of a single partial digest. Thus only one partial digest should usually suffice for reconstruction.

The combinatorial results of [18] assume that the exact length of each fragment is presented, an entirely unrealistic assumption for experimental data. In practice, the lengths of DNA fragments are measured by means of manipulating diffusion speeds in gel electrophoresis. The observed length of each fragment may have significant relative error. Currently, 2% to 5% relative error is achievable [6]. Further, two fragments of approximately equal length are unlikely to be discriminated on the gel, so the multiplicities of distances are almost certain to be lost. Additional complications arise with small fragments, which can run off the end of the gel, and so can be lost. Finally, on any particular molecule certain sites are less likely to be cut than other sites, so certain fragments might not occur with sufficient frequency to be measured.

Despite these complications, we believe that the combinatorial constraints inherent in partial digest data are sufficiently strong to facilitate reconstruction. To explore this idea, we have generalized the backtracking algorithm of [18] to deal with noisy data, both imprecise measurements as well as missing fragments. In this paper, we present our generalized algorithm for noisy data, analysis of its performance, as well as results of experiments to assess its behavior.

We emphasize that the practical instances of this problem are quite small. With more than 5-10 restriction sites, it becomes increasing difficult to distinguish the resulting fragments with electrophoresis. The time complexity of our algorithms is much less important than minimizing the number of possible interpretations for the molecule, given the number of experiments. By using partial digests, even with missing fragments, our methods perform extremely well.

In Section 2, we discuss related work, both previous algorithms for restriction site analysis and the combinatorial results which support the partial digest approach. In Section 3, we present a backtracking algorithm for reconstructing sets from noisy data and missing fragments. We analyze the performance of this algorithm on noisy data in Section 4, establishing a relative error limit of $r = \Theta(1/n^2)$ where our technique becomes infeasible. The results of experiments with an implementation of this algorithm are presented in Section 5.

2 Previous Work

Determining the location of the sites from restriction site data is a difficult algorithmic problem, which has been extensively studied. Stefik [20] approaches it as an AI problem and develops an expert system to solve it. Allison and Yee [2] develop inequalities from the data and apply separation theory to manipulate them. This approach has recently been extended to mapping three enzymes [1]. Dix and Kieronska [7] describe a program which uses backtracking to prune the possible configurations. Bellon [3] uses an approach from the theory of partitions and Tuffery, et. al [22] attempts something akin to dynamic programming. Grigorjev and Mironov [9] use a simulated annealing algorithm based on varying the lengths of the fragments. Finally, Rhee [16] gives a lucid survey of the problem and presents yet another approach, based on string matching.

These approaches attempt reconstruction by using multiple complete digests, as opposed to the single enzyme method proposed in this paper. Fragment lengths from incomplete digests have been exploited by Sutherland and Partis [21] for the related problem of mapping antibody binding sites on a protein. They use linear programming techniques as opposed to our combinatorial search methods.

Our approach is based on the theory of homeometric sets [17]. The problem of reconstructing sets from interpoint distances dates back to the origins of X-ray crystallography in the 1930's [14, 15]. Two noncongruent n-point sets are *homeometric* if the multisets of $\binom{n}{2} = n(n-1)/2$ distances they determine are the same.

We define the generating function $P(x) = \sum_i x^{a_i}$ for a given set of points $\{a_1, \ldots, a_n\}$. There is a simple correspondence between the distance multiset $d_{ij} = |a_i - a_j|$, for all $1 \leq i < j \leq n$ and the difference generating function $Q(x) = P(x) \cdot P(1/x)$. Any two sets with the same difference generating functions must be homeometric.

This generating function representation suggests the following algorithm for reconstruction [17]. Construct the polynomial $Q(x)$ and factor it into irreducibles over the ring of polynomials with integer coefficients [11], then distinguish between the factors of $P(x)$ and $P(1/x)$. Lemke and Werman [10] show that this algorithm takes time polynomial in the length of the longest distance d_{1n}, as opposed to the number of points n.

Let the function $H(n)$ denote the maximum possible number of mutually noncongruent and homeometric n-point sets which can exist in one dimension. Using combinatorial results on the number of factors of certain polynomials, [18] show that

$$\frac{1}{2} \cdot n^{0.8107144} \leq H(n) \leq \frac{1}{2} \cdot n^{1.2324827}$$

Further, $H(n)$ is always a power of 2. Therefore the maximum number of possible reconstructions for any distance set is always small, and except for pathological examples is usually one. This fact demonstrates that incomplete digests dramatically reduce the number of interpretations of a complete digest.

Newberg and Naor [13] consider a related combinatorial problem, associated with probed partial digestion mapping. Let S be a multiset of nonnegative integers, containing exactly one zero. The reconstruction problem seeks all unordered pairs $\{X, Y\}$ such that $S = X + Y = \{x + y : x \in X, y \in Y\}$. They prove that the number of solutions for a given set S can be $\Omega(N^{1.73})$, where $N = |S|$. In contrast, Goldstein and Waterman [8] showed that the number of solutions to a double digest problem can be exponential in the length of the segment.

3 The Reconstruction Algorithm

The polynomial factorization approach does not generalize to noisy data, since it would require a concept analogous to an approximate factorization. However, the reconstruction problem has a combinatorial flavor which is not captured by this approach. Skiena, Smith, and Lemke [18] presented a backtracking algorithm which, although it takes $\Theta(2^n \cdot n \log n)$ time in the worst-case [24], it generalizes readily into a practical algorithm for noisy data.

Here we review the backtracking algorithm. This presentation is taken from [18], although an alternate exposition appears in [23], where it is used to illustrate backtracking.

Let d_{ij} represent the distance between the ith and jth points on a line, ordered from left to right. The complete set of distances and their initially unknown assignments can be represented as a triangular matrix or *pyramid.* The set of distances d_{ij} such that $j-i=l$ defines row l of the pyramid. For convenience, we shall represent these distances by a vector v sorted in increasing order, so that $v[i] \leq v[j]$, $1 \leq i < j \leq \binom{n}{2}$. A solution is an assignment of all input distances to distinct positions of the pyramid such that all triangle equalities are satisfied. An input fragment is called a *base* fragment if it goes into the first row of the pyramid, and is called an *intermediate* fragment otherwise. The positions d_{1i}, $1 < i \leq n$ are called the *left* hand side of the pyramid and the positions d_{jn} are called the *right* hand side of the pyramid.

The backtracking procedure repeatedly positions the *largest remaining* unpositioned distance. Because we are placing the largest available distance in the pyramid, there will be only two possible locations to choose from. The key to making this procedure efficient is to immediately fill in the pyramid any values which are determined by our previous (non-deterministic) selections.

Assume that we have thus far placed l elements along the left-hand side of the pyramid and r on the right, as shown in Figure 2. All the positions in the shaded regions are determined by the $l+r-1$ distances d_{1j} and d_{in}, $(n-l+1) \leq j \leq n$, $1 \leq i \leq r$. The largest remaining distance must be associated with either $d_{1(n-l)}$ or $d_{(r+1)n}$.

Suppose that the backtrack procedure selects the left side to receive the next element. Thus $d_{1(n-l)}$ is assigned the largest remaining distance. This determines $d_{i(n-l)}$, $2 \leq i \leq r$, since $d_{i(n-l)} = d_{1(n-l)} - d_{1i}$, which are already in the pyramid, as well as $d_{(n-l+1)(n-j)}$, $0 \leq j \leq l-2$, since $d_{(n-l+1)(n-j)} = d_{1n} - d_{1(n-l+1)} - d_{(n-j)n}$. If any of these $l+r-1$ determined values is not an available distance, the partial reconstruction cannot be extended into a pyramid and so we backtrack. If all correspond to unused distances, we mark them as used and advance to the next level. Since the ith choice determines i values, $n-1$ choices are sufficient to determine any pyramid. On the ith choice, we will perform i binary searches in a sorted list of the distances to test whether the ones we need are available.

By assigning $d_{1n} = v[\binom{n}{2}]$, $d_{1(n-1)} = v[\binom{n}{2}-1]$, and $d_{(n-1)n} = d_{1n} - d_{1(n-1)}$, the search is initialized with $l=2$ and $r=1$.

When the data contains noise, we must use interval arithmetic to correctly handle fragments subject to a given relative error r. Now each fragment f defines an interval

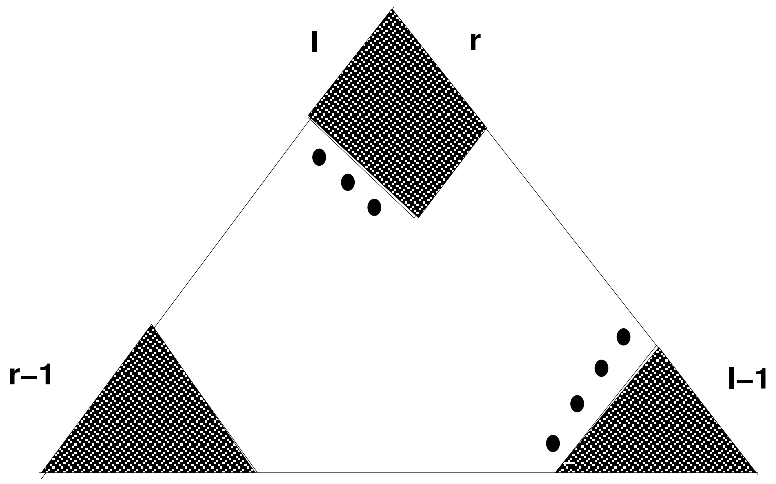

Figure 2: The Effect of Choosing $d_{1(n-l+1)}$.

$[f \cdot (1-r), f \cdot (1+r)]$. An interval is also associated with each position in the pyramid, which gives upper and lower bounds on the length of any fragment which can be placed in this slot. A fragment f can be assigned to the position d_{ij}, if the intersection of the interval corresponding to d_{ij} and $[f \cdot (1-r), f \cdot (1+r)]$ is non-empty. Define max and min as follows:

$$\max(d_{i(i+1)}) = d_{i(i+1)}(1+r), \max(d_{ik}) = \sum_{j=1}^{k-1} \max(d_{j(j+1)})$$

$$\min(d_{i(i+1)}) = d_{i(i+1)}(1-r), \min(d_{ik}) = \sum_{j=1}^{k-1} \min(d_{j(j+1)})$$

Any assignment of input fragments to the pyramid which satisfies the following constraints for all appropriate i, j, k is called *globally consistent*:

$$\max(d_{ik}) + \max(d_{kj}) \geq d_{ij} \geq \min(d_{ik}) + \min(d_{kj})$$

$$\max(d_{1n}) - \min(d_{1i}) - \min(d_{jn}) \geq d_{ij} \geq \min(d_{1n}) - \max(d_{1i}) - \max(d_{jn})$$

For d_{1l} or d_{rn} the largest remaining fragment is assigned. Our implementation checks only local interval constraints namely

$$\max(d_{i(j-1)}) + \max(d_{(j-1)j}) \geq d_{ij} \geq \min(d_{i(j-1)}) + \min(d_{(j-1)j})$$

$$\max(d_{i(i+1)}) + \max(d_{(i+1)j}) \geq d_{ij} \geq \min(d_{i(i+1)}) + \min(d_{(i+1)j})$$

during search, but then verifies the global consistency of any completed pyramid in $O(n^3)$ time.

Although there still exist only two possible positions for the largest fragment, the derived pyramid values are no longer unique, since multiple input fragments may lie in any given interval. To ensure that no solution is missed, we must continue the search with every possible combination of candidate fragments. Thus the nodes of the backtrack tree may have exponentially large outdegree, and the worst-case behavior of the algorithm may become significantly worse than $\Theta(2^n \cdot n \ln n)$. This suggests using a heuristic to select the fragment when there are multiple choices in a given interval, such as using the smallest or median fragment. Although this eliminates a potentially vast amount of redundant computation, often no solution will be found even when one exists.

A better approach to reducing redundant computation is *grouping*. The input data typically contains several pairs of fragments of such similar length that they are essentially interchangeable. Since k pairs of interchangable elements increases the number of solutions (and execution time) by 2^k, in practice the data is preprocessed by replacing all fragments whose length are within a given grouping percentage of each other with their mean. For a given sorted input, the groups can be defined in a greedy manner, so by definition an input fragment belongs to only one group. All elements in a group is replaced by the mean value of the group. Thus grouping is a preprocessing step, and we then run the algorithm on the modified input. Grouping provides a significant speedup, at the cost of potential missed solutions. The grouping idea is due to Chang [5]. In a 1989 empirical study of probed partial digestion mapping, Chang grouped fragments of similar length to reduce the number of solutions.

4 Analysis

Although the backtracking algorithm takes exponential time in the worst case, we expect better behavior in practice, since the search is pruned whenever one of the $i-1$ derived values determined on the ith level does not appear in the set of remaining distances. Our experiments have shown the algorithm is capable of handling distance sets of several hundred points, when the fragments are presented without error. The reason is that when the distance set arises from n real points in general position, i.e., no two points coincide, the incorrect position for the largest distance will (with probability 1) be pruned immediately, and so the procedure uses $\Theta(n^2 \log n)$ operations. In this section, we analyze the sensitivity of this algorithm to the relative error r in the input, although the algorithm can be modified to handle different error levels for different fragments.

To analyze the behavior of our reconstruction algorithm, we assume that fragment lengths are generated according to the following probabilistic model. Let $[a, b]$ denote the closed interval from a to b. In the interval $[1, M]$ a set of integer points S are chosen according to a binomial distribution $B(M, p)$, where $p = n/M$. Thus the expected number of points in S is n. If we assume that DNA molecules are random strings, then $p \approx 1/4^k$ for a restriction enzyme which cuts at a pattern of length k.

In the backtracking procedure, the act of placing the kth largest fragment defines an interval for the value of $k-1$ positions in pyramid. If any one of these intervals does not contain an available element of the input distance set, we immediately backtrack. If an interval contains multiple fragments, we must continue the search with all combinations of the multiple fragments. Thus there are two critical points in the algorithm; first, where the probability of a false match becomes high enough that the procedure backtracks infrequently, and second where intervals contain multiple fragments often enough that each tree node has high outdegree. By arbitrarily selecting one of the multiple fragments in the interval, using a pre-selection rule such as the median, we avoid the effects of this second phenomenon.

Theorem 1 *Let I be the set of fragments resulting from selecting points from the interval $[1, M]$ according to a binomial distribution, with $p = n/M$, where $M \gg n$ (ie. such that $\sum_{i=1}^{\infty} \frac{n^i}{i \cdot M^i} = \frac{n}{M}(1+o(1))$). Assume that all base fragments in I are identified, and that each fragment in I is measured with a relative error of at most r. Then the expected time of a pre-selection algorithm on I is $O(n^3)$ when $r \leq \ln 2/(2n)$.*

Proof: The kth step of the backtrack algorithm consists of positioning the largest remaining fragment either left or right, and then checking whether there exist unused fragments of I for each of the k derived intervals. If any one of these intervals is empty, we backtrack. Exactly one of these derived intervals corresponds to a base fragment, and there are only n candidates to fill it since the rest of the $\binom{n}{2}$ fragments are marked intermediate. Thus the probability $P(w)$ that the base interval is empty presents a lower bound on the probability of backtracking on any false branch, since we can possibly backtrack on any intermediate interval as well.

Let S_i denote the base fragment defined between the $(i-1)$st and ith selected points of I. Let A_i denote the event that $|S_i| \in [w - rw, w + rw]$, for some given w. Then

$$P[w] = P[\cap_{i=1}^{n} \bar{A}_i] = P[\bar{A}_1]^n$$

since all events are mutually independent and identically distributed. Further, for a given w,

$$\begin{aligned} P[\bar{A}_1] &= 1 - P[A_1] \\ &= 1 - (1-p)^{w(1-r)} \sum_{i=0}^{2rw-1} p(1-p)^i \\ &= 1 - (1-p)^{w(1-r)}(1 - (1-p)^{2rw}) \end{aligned}$$

To obtain a lower bound on the backtrack probability, we must find which w minimizes $P[\bar{A}_1]$. Let $q = (1 - n/m)$. Then

$$\begin{aligned} \frac{d\, P[\bar{A}_1]}{dw} &= -q^{w(1-r)} \cdot \ln(q^{1-r}) + q^{w(1+r)} \cdot \ln(q^{1+r}) \\ &= q^w \ln(q)(q^{rw}(1+r) - q^{-rw}(1-r)) \end{aligned}$$

Setting the derivative to zero gives $(1 - n/M)^{2rw} = (1-r)/(1+r)$. Taking the logarithm of both sides gives

$$\begin{aligned} w &= \frac{1}{2r}[\frac{\lg(1-r) - \lg(1+r)}{\lg((1-n/M))}] \\ &= \frac{1}{2r}[\frac{(-r - r^2/2 - r^3/3 - \ldots) - (r - r^2/2 + r^3/3 - \ldots)}{-n/M - n^2/(2M^2) - \ldots}] \end{aligned}$$

Since $r \ll 1$ and $p \ll 1$, $P(w)$ is minimized for $w \approx M/n$.

In the backtrack tree corresponding to a particular execution of the algorithm, there will be one path of length n from the root corresponding to the correct reconstruction. Hanging off each node on this path will be a subtree, the size of which depends upon how often unused fragments happen to satisfy the derived intervals. We can model this with the following process. Let p' be the probability that a node dies without children, while it has exactly two children with probability $1-p'$. The expected number of nodes produced by this process is given by $X[p'] = p' + 2(1-p')X[p']$, so $X[p'] = p'/(2p'-1)$. Therefore all false subtrees will be of constant size whenever $P(w) > 1/2$ for all w.

To determine r such that $P(w) > 1/2$, using $w = M/n$ to minimize $P(w)$ gives

$$(1-(1-n/M)^{w(1-r)}(1-(1-n/M)^{2rw}))^n > \frac{1}{2}$$

Since $\lim_{n\to\infty}(1-1/n)^n \to 1/e$,

$$(1-e^{-(1-r)}(1-e^{-2r}))^n > \frac{1}{2}$$

and

$$1-e^{-2r} < e^{(1-r)}(1-(1/2)^{1/n})$$

. We show that the above inequality is true whenever $r < \ln 2/(2n)$. This implies that $r < 1$ and $e^{1-r} > 1$, so the result follows if $1-e^{-2r} > 1-1/2^{1/n}$, which is true when $r < \ln 2/(2n)$. ∎

The heuristic algorithm analyzed in Theorem 1 avoids the effects of multiple fragments per interval, at the cost of possibly missing solutions. For a more complete analysis of the backtracking algorithm, we need two quantities: the probability that an interval is devoid of fragments, and the probability that an interval does not contain multiple fragments.

Recall that S is the set of randomly chosen integer points. Following the notation of [19], fixing w and r, $x, y \in S$ is said to form a *bond* $x \sim y$ if $|x-y| \in I = [w\cdot(1-r), w\cdot(1+r)]$. The points x, y, z form a *vee* if $x \sim y$ and $x \sim z$, denoted $x, \{y, z\}$.

Let $N = N(M,w,r)$ denote the potential number of bonds and $V = V(M,w,r)$ denote the potential number of vees, for the given values of M, w, and r.

Lemma 1 $N(M,w,r) = (M-w)(2rw+1)$.

Proof: We count the bonds $x \sim y$, such that $x < y$, $1 \le x \le M$. For any $x \le M-w(1+r)$, all $2rw+1$ possible bonds exist. Edge effects reduce this number for other values of x. When $x > M-w(1-r)$, no bonds exist. When $M-w(1-r) \ge x > M-w(1+r)$, there are exactly $M-w(1-r)+1-x$ bonds. Thus

$$N(M,w,r) = (2rw+1)(M-w(1+r)) + \sum_{i=1}^{2rw} i = (M-w)(2rw+1)$$

∎

Lemma 2 $V(M,w,r) \leq 2r^2w^2M(1+o(1))$.

Proof: We count the vees $x,\{y,z\}$, such that $1 \leq x \leq M$. There are three cases, depending upon the value of w. For a point x, we denote the right window as $[x+w(1-r), x+w(1+r)]$ and the left window $[x-w(1+r), x-w(1-r)]$. The situation is complicated by windows which partially intersect $[1,M]$.

Case 1. $w \leq M/2 - rw$: When the left window is not entirely contained in $[1,M]$, the right window is entirely contained in $[1,M]$ and vice versa. We partition $[1,M]$ according to the position of x:

Case 1a. $1+(w+rw) \leq x \leq M-(w+rw)$: The number of vees associated with x is $\binom{4rw+2}{2}$ and the number of such x's are $M-2(w+rw)$. The case of a partial left window is identical to the case of partial right window, hence the number of such vees is $2(M-2(w+rw))\binom{4rw+2}{2}$.

Case 1b. $w-rw < x \leq w+rw$: The number of vees for such x is

$$\sum_{i=1}^{2rw} \binom{i}{2} + i(2rw+1) + \binom{2rw+1}{2} = (4rw+1)\binom{2rw+1}{2} + \binom{2rw+1}{3}$$

Case 1c. $x \leq 1+(w-rw)$: The number of vees for each x is $\binom{2rw+1}{2}$. Since there are $w-rw$ such x, the number of such vees is $(w-rw)\binom{2rw+1}{2}$.

In summary, for $w \leq M/2(1+r)$,

$$V \approx (M-2w) + rw(6M-10w) + r^2w^2(8M-12w) - 25r^3w^3$$

Case 2. $M/2+rw \geq w > M/2-rw$: For any point x, the length of window contained in $[1,M]$ is $(2rw+1)$, hence there are $M\binom{2rw+1}{2} = Mrw(2rw+1) \approx 2Mr^2w^2$ vees.

Case 3. $(w-rw) > M/2$: For any point x, either the left or right window (but not both) is contained in $[1,M]$. The number of such vees is

$$2(M-w-rw)\binom{2rw+1}{2} + 2\binom{2rw+1}{3} \approx 2rw(M-w) + 4r^2w^2(M-w) - \frac{4}{3}r^3w^3$$

Since $rw \gg 1$, the dominant term in all three cases is r^2w^2M. Further, the maximum value of these terms (from Case 1) is $2r^2w^2M$. ∎

Now, we compute the probability of zero bonds for a given interval. Let the event A_i, $1 \leq i \leq N$ denote that the ith bond is present in S. Hence $P[A_i] = p^2$. The expected number of bonds is $N \cdot p^2$ and the expected number of vees $\Delta = V \cdot p^3$. Let Z denote the number of bonds. Then by Janson's inequality [4]

$$\prod_{i=1}^{N} P[\bar{A}_i] \leq P[Z=0] \leq \prod_{i=1}^{N} P[\bar{A}_i] \cdot e^{\Delta/(1-p^2)} \tag{1}$$

Since $p \ll 1$ in our application, the upper bound can be taken as $\prod P[A_i] \cdot e^{\Delta}$. Also, since $p^2 \ll 1$, $1-p^2 \approx e^{-p^2}$ and (1) can be written as

$$e^{-Np^2} \leq P[Z=0] \leq e^{-Np^2+\Delta} \tag{2}$$

Lemma 3 *$P[Z>1] \leq (r^2n^4/8)\cdot(1+o(1))$ for all $w \in [1,M]$.*

Proof: Since

$$\sum_{i=2}^{\infty} P[Z=i] \leq \sum_{i=0}^{\infty} i \cdot P[Z=i] - \sum_{i=1}^{\infty} P[Z=i]$$

we know $P[Z>1] \leq E[Z] - P[Z>0]$. ¿From Janson's inequality, $P[Z=0] \leq e^{-Np^2+\Delta}$ and hence $P[Z>0] \geq 1 - e^{-Np^2+\Delta}$. Therefore,

$$\begin{aligned} P[Z>1] &\leq E[Z] - (1 - e^{-(Np^2-\Delta)}) \\ &= Np^2 - ((Np^2-\Delta) - (Np^2-\Delta)^2/2! + \cdots) \\ &\leq \Delta + (Np^2-\Delta)^2/2 \end{aligned}$$

Since $Np^2 > \Delta$ when $rn < 1$, $P[Z>1] \leq (Np^2)^2/2 + \Delta$. By Lemma 1, $N = 2rw(M-w)$. This is maximized when $w = M/2$, so $N \leq rM^2/2$ and $Np^2 \leq rn^2/2$. We seek an upper bound on Δ. By Lemma 2, $V \leq c \cdot r^2 w^2 M$ for some positive constant c. By definition, $\Delta = Vp^3$, so $\Delta \leq c \cdot r^2 n^3$. Since $Np^2 \leq rn^2/2$,

$$P[Z>1] \leq r^2n^4/8 + c \cdot r^2 n^3,$$

giving the result. ∎

Theorem 2 *Let I be the set of fragments generated by selecting points from an interval of length M according to a binomial distribution, with $p = n/M$, where $M \gg n$ (ie. such that $\sum_{i=1}^{\infty} \frac{n^i}{i \cdot M^i} = \frac{n}{M}(1+o(1))$). Assume that all base fragments in I are identified, and that each fragment in I measured with a relative error of at most r. Then the expected time of the backtracking algorithm on I is $O(n^3)$ when $r = o(1/n^2)$.*

Proof: By Lemma 3, for $r = o(1/n^2)$, the probability of an interval containing multiple fragments becomes vanishingly small. Thus the performance of the algorithm is governed by the probability of false matches, as in the median selection algorithm. Theorem 1 shows that false matches cannot lead to expected exponential time until $r > \ln 2/(2n)$, thus giving the result. ∎

5 Experimental Results

We conducted our experiments on the bacteriophage λ [12], a common cloning vehical with seven restriction sites for the enzyme *Hind*III. The distance between adjacent sites on this molecule are $\{23130, 2027, 2322, 9416, 564, 125, 6557, 4361\}$ which leads to the following set of 36 pairwise distances:

```
125, 564, 689, 2027, 2322, 4349, 4361, 6557, 6682, 7246, 9416, 9980,
10105, 10918, 11043, 11607, 11738, 12302, 12427, 13765, 14329, 14454,
16662, 18984, 21011, 21023, 23130, 23345, 25157, 25372, 27479, 36895,
37459, 37584, 44141, 48502
```

We perturbed the length of each of these fragments to simulate the effects of experimental error on the performance of our algorithm. Specifically, to simulate a relative error r, each fragment of length f was replaced with a random integer selected uniformly from the interval $[f \cdot (1-r), f \cdot (1+r)]$.

Our results are summarized in a series of tables. Each problem instance has an associated relative error r, and was run under a variety of different grouping factors. For a given grouping factor g, any fragments of length $l_1 \leq \ldots \leq l_k$ are replaced by $\sum_{i=1}^{k} l_i/k$ if $l_k \leq (1+g) \cdot l_1$. Increasing the grouping factor has the effect of dramatically reducing the search time while less dramatically reducing the number of solutions. For each pair of r and g, the execution time and number of solutions have been reported. An '∞' denotes pairs where the program was not run to completion.

Table 1 illustrates the performance of our algorithm when all 36 fragments have been measured with a given maximum relative error. All experiments were performed on a Sun Sparcstation 2, and no results requiring over five minutes of user time are reported. Given all fragments, we can efficiently solve problem instances with up to 5.0% relative error for certain grouping factors. Although the number of solutions increases as high as 16 (for $r = 3.0\%$, $g = 0.0\%$), inspection of the actual solutions reveals that they are only minor perturbations of each other:

$$\begin{array}{ll}
\{22768, 2062, 2323, 9394, 576, 126, 6737, 4330\}, & \{22841, 2062, 2323, 9394, 576, 126, 6737, 4330\} \\
\{22768, 2323, 2062, 9394, 576, 126, 6737, 4330\}, & \{22841, 2323, 2062, 9394, 576, 126, 6737, 4330\} \\
\{22768, 2062, 2323, 9394, 576, 126, 6718, 4330\}, & \{22841, 2062, 2323, 9394, 576, 126, 6718, 4330\} \\
\{22768, 2323, 2062, 9394, 576, 126, 6718, 4330\}, & \{22841, 2323, 2062, 9394, 576, 126, 6718, 4330\} \\
\{22768, 2062, 2323, 9394, 576, 126, 6737, 4229\}, & \{22841, 2062, 2323, 9394, 576, 126, 6737, 4229\} \\
\{22768, 2323, 2062, 9394, 576, 126, 6737, 4229\}, & \{22841, 2323, 2062, 9394, 576, 126, 6737, 4229\} \\
\{22768, 2062, 2323, 9394, 576, 126, 6718, 4229\}, & \{22841, 2062, 2323, 9394, 576, 126, 6718, 4229\} \\
\{22768, 2323, 2062, 9394, 576, 126, 6718, 4229\}, & \{22841, 2323, 2062, 9394, 576, 126, 6718, 4229\}
\end{array}$$

We find no evidence that multiple interpretations present a significant problem in our approach. Indeed, none of our experiments to date have resulted in two significantly different interpretations of the same data.

By performing both a complete and an incomplete digest on clones of a given molecule, the base fragments of our pyramid can be instantly identified, reducing the search times substantially for a given problem instance. Table 2 uses the same fragment lengths as Table 1, but the base fragments have been identified to reduce the search. Under this model, problem instances with relative errors of up to 7.0% can be solved.

Missing fragments are a fact of life in experimental data, and present a significant challenge to our algorithm. The position of the wildcard entry can have a significant effect on execution time. In Table 3, one intermediate fragment has been removed,

Reconstruction Time in Seconds

rel. error	*grouping percentage* 0.0%	0.5%	1.0%	1.5%	2.0%	2.5%
0.0%	0.04					
0.5%	0.07					
1.0%	0.1	0.05				
1.5%	0.97	0.12				
2.0%	3.32	0.65	0.05			
2.5%	9.12	3.46	0.15			
3.0%	97.37	37.6	15.31	4.8		
3.5%	∞	∞	74.34	36.16		
4.0%	∞	∞	∞	87.9	54.63	
4.5%	∞	∞	∞	∞	78.75	
5.0%	∞	∞	∞	∞	∞	59.53

Number of Solutions

rel. error	*grouping percentage* 0.0%	0.5%	1.0%	1.5%	2.0%	2.5%
0.0%	1					
0.5%	3					
1.0%	4	2				
1.5%	4	4				
2.0%	8	4	0			
2.5%	8	8	0			
3.0%	16	8	4	4		
3.5%	∞	∞	8	6		
4.0%	∞	∞	∞	6	6	
4.5%	∞	∞	∞	∞	0	
5.0%	∞	∞	∞	∞	∞	0

Table 1: Performance when provided with 36 of 36 fragments, with none of the segments identified.

Reconstruction Time in Seconds

	grouping percentage						
rel. error	0.0%	0.5%	1.0%	1.5%	2.0%	2.5%	3.0%
0.0%	0.05						
0.5%	0.03						
1.0%	0.04	0.06					
1.5%	0.35	0.05					
2.0%	0.53	0.22	0.03				
2.5%	1.81	0.63	0.07				
3.0%	5.55	3.03	1.51	1.47			
3.5%	7.66	2.18	0.73	0.38			
4.0%	24.17	7.0	1.39	0.24	0.16		
4.5%	67.31	35.76	20.62	1.27	0.27		
5.0%	∞	172.41	30.3	7.01	0.88	0.38	
5.5%	∞	∞	108.72	5.42	5.41	0.86	
6.0%	∞	∞	114.45	109.74	32.51	11.63	4.87
6.5%	∞	∞	∞	∞	161.19	99.1	14.76
7.0%	∞	∞	∞	∞	265.12	266.83	78.58

Number of Solutions

	grouping percentage						
rel. error	0.0%	0.5%	1.0%	1.5%	2.0%	2.5%	3.0%
0.0%	1						
0.5%	1						
1.0%	1	1					
1.5%	1	1					
2.0%	1	1	0				
2.5%	1	1	0				
3.0%	2	1	1	1			
3.5%	2	2	2	1			
4.0%	2	2	2	1	1		
4.5%	2	2	2	2	0		
5.0%	∞	3	2	2	0	0	
5.5%	∞	∞	3	1	1	0	
6.0%	∞	∞	2	2	0	0	
6.5%	∞	∞	∞	∞	0	0	0
7.0%	∞	∞	∞	∞	3	3	0

Table 2: Performance when provided with 36 of 36 fragments, with all of the base fragments identified.

Reconstruction Time in Seconds

rel. error	*grouping percentage* 0.0%	0.5%	1.0%	1.5%	2.0%	2.5%
0.0%	0.03					
0.5%	0.04					
1.0%	0.09	0.06				
1.5%	0.43	0.09				
2.0%	1.21	0.65	0.45			
2.5%	1.86	0.75	0.09			
3.0%	16.26	9.17	3.55	4.01		
3.5%	31.46	10.95	3.4	1.43		
4.0%	138.91	36.4	12.35	1.49	1.12	
4.5%	∞	∞	207.74	16.08	3.18	
5.0%	∞	∞	281.88	81.51	14.72	6.98
5.5%	∞	∞	∞	56.74	57.47	15.33

Number of Solutions

rel. error	*grouping percentage* 0.0%	0.5%	1.0%	1.5%	2.0%	2.5%
0.0%	1					
0.5%	1					
1.0%	1	1				
1.5%	1	1				
2.0%	2	2	0			
2.5%	1	1	0			
3.0%	2	2	2	2		
3.5%	2	2	2	1		
4.0%	2	2	2	1	1	
4.5%	∞	∞	2	2	0	
5.0%	∞	∞	2	2	0	0
5.5%	∞	∞	∞	1	1	0

Table 3: Performance when provided with 35 of 36 fragments. One intermediate segment is missing, and all base segments identified.

Number of Solutions

rel. error	*grouping percentage* 0.0%	0.5%	1.0%
0.0%	0.2		
0.5%	0.4		
1.0%	0.94	1.1	
1.5%	4.1	2.6	
2.0%	11.2	12.4	3.9
2.5%	58.5	117.1	17.7

Number of Solutions

rel. error	*grouping percentage* 0.0%	0.5%	1.0%
0.0%	1		
0.5%	1		
1.0%	1	1	
1.5%	1	1	
2.0%	2	2	2
2.5%	1	1	1

Table 4: Performance when provided with 33 of 36 fragments. One base fragment and two intermediate fragments are missing, with all remaining labeled.

Reconstruction Time in Seconds

	grouping percentage			
rel. error	0.0%	0.5%	1.0%	1.5%
0.0%	0.03 (0)			
0.5%	0.04 (0)			
1.0%	0.05 (0)	0.1 (3)		
1.5%	0.26 (0)	0.13 (5)		
2.0%	0.32 (0)	3.13 (5)	0.29 (11)	
2.5%	1.08 (0)	14.52 (3)	0.77 (8)	
3.0%	2.92 (0)	∞ (4)	239.41 (6)	55.31 (7)
3.5%	4.04 (0)	23.59 (2)	240.63 (6)	∞ (8)
4.0%	13.34 (0)	65.59 (2)	∞ (6)	∞ (9)
4.5%	37.01 (0)	133.01 (1)	∞ (4)	∞ (8)

Number of Solutions

	grouping percentage			
rel. error	0.0%	0.5%	1.0%	1.5%
0.0%	1			
0.5%	1			
1.0%	1	1		
1.5%	1	1		
2.0%	1	1	0	
2.5%	1	2	1	
3.0%	2	∞	3	0
3.5%	2	2	2	∞
4.0%	2	2	∞	∞
4.5%	2	2	∞	∞

Table 5: Performance when provided only with non-identical fragments, with all other segments identified.

Reconstruction Time in Seconds

N	0.0%	0.1%	0.2%	0.3%	0.4%	0.5%	1.0%	1.5%	2.0%	2.5%
10	.02-.03	.03-.05	.01-.15	0-.36	0-.9	0-1.1	0-7.7	0-25.5	0-152	0.2-38.8
11	.23-.49	.01-.11	0-.35	0-.62	0-2.04	0-9.7	0-140	0-91.1	100-196	258-274
12	.02-.06	.02-.05	.05-.08	.08-.14	.13-.96	0-4.22	10.6-131	∞	∞	∞
13	.03-.05	.03-.06	.04-.16	.12-.58	0-5.4	2-14.3	∞	∞	∞	∞
14	.03-.05	.12-.19	.01-.94	.09-2.7	0-40.5	0-127	∞	∞	∞	∞
15	.03-.08	1-1.4	1.8-2.7	1.3-1.5	12.9-16.6	78.9-93.3	∞	∞	∞	∞
16	.03-.06	.08-.26	.3-12.1	.62-51.6	12.5-18.9	75.9-94.9	∞	∞	∞	∞
17	.05-.07	0-13.9	0-10.4	3.4-10.8	31.9-49.4	142.4-153.8	∞	∞	∞	∞
18	.05-.08	5.5-77.6	23.7-84.9	162.1-169.5	∞	∞	∞	∞	∞	∞
19	.05-.08	9.5-84.9	125-139	∞	∞	∞	∞	∞	∞	∞
20	.05-.08	79.4-123.6	88.1-114.5	∞	∞	∞	∞	∞	∞	∞

Table 6: Performance on randomly generated input, with all base and intermediate segments identified. Confidence interval for μ based on t-dn with $\alpha = 0.05$ and $n = 5$.

while for Table 4 three fragments have been removed (one of which is a base fragment). Even under these conditions, relative errors of up to 2.5% can be processed.

Perhaps the most common cause of missing fragments occurs when two fragments of similar length are not resolved on the gel. In such a case, the approximate length of the fragments are known, but not the multiplicity. For the experiments reported in Table 5, all fragments clustered together within the given grouping percentage were replaced by one instance of the mean length of these fragments. Thus, unlike our other tables, the problems get harder with increasing grouping percentage. The number of missing fragments is reported within parentheses in Table 5. All missing fragments were assumed to be multiple occurrences of existing fragments instead of unrestricted wildcards. Problems with as many as eight missing fragments are successfully handled, and the number of satisfying solutions remains extremely small.

Our final experiment concerned the effect of relative error on random instances of different sizes. Five random instances of N base fragments were generated and subjected to a random relative error of r, where $10 \leq N \leq 20$ and $0.0\% \leq r \leq 2.5\%$. Table 5 summarizes these experiments, by presenting the confidence interval with $\alpha = 0.05$ for the average running time in seconds. It clearly illustrates that the critical region is $r = \Theta(1/n^2)$, as is predicted by the theory.

The conclusion to be drawn from these experiments is that our technique of restriction site analysis via partial digests successfully reconstructed simulated data, under realistic error tolerances. We are currently seeking actual experimental data as the next stage in this research.

6 Acknowledgements

We would like to thank Bill Chang, Scott Davey, and Tom Marr of Cold Spring Harbor Laboratory for supplying data and for their interest in this work. Discussions with Joel Spencer were helpful with the probabilistic analysis. We thank the referees for comments which improved the presentation of these results.

References

[1] L. Allison, S. Ho, and C. N. Yee. Restriction site mapping for three or more enzymes. *Comput. Appl. Biol. Sci.*, 6:195–204, 1990.

[2] L. Allison and C. N. Yee. Restriction site mapping is in separation theory. *Comput. Appl. Biol. Sci.*, 4:97–101, 1988.

[3] B. Bellon. Construction of restriction maps. *Comput. Appl. Biol. Sci.*, 4:111–115, 1988.

[4] R. Boppana and J. Spencer. A useful elementary correlation inequality. *Journal of Combinatorial Theorey, Series A*, 50:305–307, 1989.

[5] W. I. Chang. personal communication, 1991.

[6] W. I. Chang and T. Marr. personal communication. Cold Spring Harbor Laboratory, April 1992.

[7] T.I. Dix and D.H. Kieronska. Errors between sites in restriction site mapping. *Comput. Appl. Biol. Sci.*, 4:117–123, 1988.

[8] L. Goldstein and M.S. Waterman. Mapping dna by stochastic relaxation. *Adv. in Applied Math.*, 8:194–207, 1987.

[9] A. V. Grigorjev and A. A. Mironov. Mapping DNA by stochastic relaxation: a new approach to fragment sizes. *Comput. Appl. Biol. Sci.*, 6:107–111, 1990.

[10] P. Lemke and M. Werman. On the complexity of inverting the autocorrelation function of a finite integer sequence, and the problem of locating n points on a line, given the unlabeled distances between them. 1988.

[11] A. K. Lenstra, H.W. Lenstra Jr., and L. Lovasz. Factoring polynomials with rational coefficients. *Math. Ann.*, 261:515–534, 1982.

[12] T. Maniatis, E. F. Fritsch, and J. Sambrook. *Molecular Cloning, a laboratory manual.* Cold Spring Harbor Laboratory, Cold Spring Harbor NY, 1982.

[13] L. Newberg and D. Naor. A lower bound on the number of solutions to the probed partial digest problem. *Advnces in Applied Math.*, 14(2), 1993.

[14] A. L. Patterson. A direct method for the determination of the components of interatomic distances in crystals. *Zeitschr. Krist.*, 90:517–542, 1935.

[15] A. L. Patterson. Ambiguities in the x-ray analysis of crystal structures. *Phys. Review*, 65:195–201, 1944.

[16] G. Rhee. DNA restriction mapping from random-clone data. Technical Report WUCS-88-18, Department of Computer Science, Washington University, St. Louis MO, 1988.

[17] J. Rosenblatt and P. Seymour. The structure of homometric sets. *SIAM J. Alg. Disc. Methods*, 3:343–350, 1982.

[18] S. S. Skiena, W. D. Smith, and P. Lemke. Reconstructing sets from interpoint distances. In *Proc. Sixth ACM Symp. Computational Geometry*, pages 332–339, 1990.

[19] J. Spencer and J. Pach. Bounds on no bonds and distance gaps in the plane, 1992.

[20] M. Stefik. Inferring DNA structures from segmentation data. *Artificial Intelligence*, 11:85–114, 1978.

[21] R. A. Sutherland and M. D. Partis. Using fragment lengths from incomplete digestion by multiply cleaving enzymes to map antibody binding sites on a protein. *Comput. Appl. Biol. Sci.*, 8:1–4, 1992.

[22] P. Tuffery, P. Dessen, C. Mugnier, and S. Hazout. Restriction map construction using a 'complete sentences compatibility' algorithm. *Comput. Appl. Biol. Sci.*, 4:103–110, 1988.

[23] M. A. Weiss. *Data Structures and Algorithm Analysis.* Benjamin Cummings, Redwood City CA, 1992.

[24] Z. Zhang. A exponential example for partial digest mapping algorithm. *Tech Report, Computer Science Dept., Penn. State Univeristy*, 1993.