Genomics via Optical Mapping II: Ordered Restriction Maps

Thomas S. Anantharaman
Bud Mishra
David C. Schwartz[1]

Abstract

In this paper, we describe our algorithmic approach to constructing ordered restriction maps based on the data created from the images of population of individual DNA molecules (clones) digested by restriction enzymes. The goal is to devise map-making algorithms capable of producing high-resolution, high-accuracy maps rapidly and in a scalable manner. The resulting software is a key component of our optical mapping automation tools and has been used routinely to map cosmid, lambda and BAC clones. The experimental results appear highly promising.

1 Genomics and Optical Mapping

Optical mapping [CAH+95, CJI+96, HRL+95, JRH+96, MBC+95, SCH+95, SLH+93, WHS95] is a single molecule methodology for the rapid production of ordered restriction maps from individual DNA molecules. Ordered restriction maps were constructed originally from yeast chromosomes by using fluorescence microscopy to visualize restriction endonuclease cutting events on individual fluorochrome-stained DNA molecules [SCH+95, SLH+93]. Restriction enzyme cleavage sites are visible as gaps that appear flanking the relaxed DNA fragments (pieces of molecules between two consecutive cleavages). Relative fluorescence intensity (measuring the amount of fluorochrome binding to the restriction fragment) or apparent length measurements (along a well-defined "backbone" spanning the restriction fragment) have proven to be accurate size-estimates of the restriction fragment and were used to construct the final restriction map. It is worth noting at this point that such a restriction map created from one single DNA molecule is limited in its accuracy by the resolution of the microscopy, the imaging system (CCD camera, quantization level, etc.), illumination and surface conditions. Furthermore, depending on the digestion rate and the noise inherent to the intensity distribution along the DNA molecule, with some probability one is likely to miss a small fraction of the restriction sites or introduce spurious sites.

[1] *Authors' Current Address: Computer Science and Chemistry Department, New York University, New York 10012.* *The research presented here was supported by* **National Center for Human Genome Research** *("New Physical Methodologies for Genomic Analysis," 2 RO1 HG00225-04, 1/1/91 through 7/15/97) and also by a grant from the* **Chiron Corporation**.

Additionally, we may sometimes (rather infrequently) lack the exact orientation information (whether the left-most restriction site is the first or the last). Thus, given two arbitrary single molecule restriction maps for the same DNA clone obtained this way, we expect them to be roughly the same in the following sense—if we "align" the maps by first choosing the orientation and then identifying the restrictions sites that differ by small amount, then most of the restrictions sites will appear roughly at the same place in both the maps.

Clearly, there are two approaches to further improve the accuracy and resolution of the maps: namely, improve the chemical and optical processes to minimize the effect of each error source and secondly, to use statistical approaches where the restriction maps of a large number of identical clones are combined to create a high-accuracy restriction map. These two approaches are not mutually exclusive and interesting trade-offs exist that can be exploited fruitfully. A large well-coordinated multidisciplinary effort at our laboratory has attacked this problem by continuously improving the chemical, optical, computational and automation aspects.

For instance, in the original method, fluorescently-labeled DNA molecules were elongated in a flow of molten agarose containing restriction endonucleases, generated between a cover-slip and a microscope slide, and the resulting cleavage events were recorded by fluorescence microscopy as time-lapse digitized images [SLH+93]. The second generation optical mapping approach, which dispensed with agarose and time-lapsed imaging, involve fixing elongated DNA molecules onto positively-charged glass surfaces, thus improving sizing precision as well as throughput for a wide range of cloning vectors (cosmid, bacteriophage, and yeast or bacterial artificial chromosomes (YAC or BAC)). Further improvements have recently come from many sources: development of a simple and reliable procedure to mount large DNA molecules with good molecular extension and minimal breakage; optimization of the surface derivatization, maximizing the range of usable restriction enzymes and retention of small fragments; and development of an open surface digestion format, facilitating access to samples and laying the foundations for automated approaches to mapping large insert clones.

The complementary sets of improvement have come from powerful statistical tools that process a preliminary collection of single-molecule restriction maps, each one created from an image of a DNA molecule belonging to a pool of identical clones. Such a collection of restriction maps are almost identical with small variations resulting from sizing errors, partially digested restriction sites and "false" restriction sites and can be combined easily in most cases. However, the underlying statistical problem poses many fundamental challenges; for example, we will show in a later section that the presence of some uncertainty in the alignment of a molecule (both orientation and/or matching in the sites) in conjunction with either false cuts or sizing error is sufficient to make the problem infeasible (NP-complete [GJ79]).(Also, see Dančík et al. [DHM97] for some related results on the complexity of a somewhat idealized model of this problem.) We note parenthetically that these negative results only correspond to pathological cases that are unlikely to occur in real life and we demonstrate that there are good probabilistic algorithms (using a Bayesian scheme) that can handle this problem adequately. Nonetheless, these negative results play an important role in clarifying the care needed in structuring the algorithm properly.

The paper is organized as follows: In section 2 and 3, we describe the restriction map model and formulate the underlying algorithmic problems. We also present several results on the worst-case complexity of the problem. In section 4, we describe a statistical model for the problem based on rather simple assumptions on the distributions of the bases in DNA and the properties of the chemical processes involved in optical mapping. These models are then used to devise probabilistic algorithms with good average time complexity. The algorithm produces as its output several maps ranked by a "quality of goodness." Additionally, it gives estimates of several auxiliary parameters governed by the underlying chemical, optical and image analysis processes (e.g., the digestion rate, false-cut rate, sizing error, contamination with other molecules, etc.). In section 5, we present experimental results on wide array of data sets (lambdaphage, cosmids; BAC data will be presented in a sequel). We conclude with a discussion of the results and future planned modifications. The relevant background material can be found in the following references: discussion on restriction maps and their role in human genome project [Kar93, KH92, Nic94, Pev90, Pri95, Wat89, Wat95, Wat77], statistics of restriction maps [LW88, Lan95a, Lan95b, Wat95] and the algorithmic and computational complexity issues [BSP+90, GGK+95, Kar93, Kra88, Lan95a, Lan95b, PW95, Wat95].

2 Restriction Map Models

Our problem can be formulated mathematically as follows. Assuming that all individual single-molecule restriction maps correspond to the same clone, and that the imaging algorithm can only provide the fragment size estimates that are scaled by some unknown scale factor, we represent a single molecule restriction map (SMRM) by a vector with ordered set of rational numbers on the open unit interval $(0,1)$:

$$D_j = (s_{1j}, s_{2j}, \ldots, s_{M_j,j}), \quad 0 < s_{1j} < s_{2j} < \cdots < s_{M_j,j} < 1, \quad s_{ij} \in \mathbb{Q}$$

By $D_j + c$ (a rational $c \in [0,1]$), we denote the vector

$$D_j + c = (s_{1j} + c, s_{2j} + c, \ldots, s_{M_j,j} + c),$$

where $-s_{1j} < c < 1 - s_{M_j,j}$.

Given a rational number $s \in (0,1)$, its reflection is denoted by $s^R = 1 - s$. Similarly, by D_j^R, we denote the vector

$$D_j^R = (s^R_{M_j,j}, \ldots, s^R_{2j}, s^R_{1j}).$$

Note that if the entries of D_j are ordered and belong to the open unit interval, so do $D_j + c$ and D_j^R provided that c is appropriately constrained.

Thus our problem can be described as follows: given a collection of data (SMRM vectors)

$$D_1, D_2, \ldots, D_m$$

we need to compute a final vector H

$$H = (h_1, h_2, \ldots, h_N)$$

such that H is "consistent" with each D_j. Thus, H represents the correct restriction map and D_j's correspond to several "corrupted versions" of H. We shall define such a general notion of "consistency" using a Bayesian approach which depends on the conditional probability that a data item D_j can be present given that the correct restriction map for this particular clone is H.

However, any such consistency requirement must satisfy certain straightforward conditions, under certain side information. For instance, if we assume that there is no false-cut and the sizing information is accurate (but the digestion may be partial), then it must be the case that for each j, either $D_j \subseteq H$ or $D_j^R \subseteq H$. In particular, if the digestion is complete (ideal case) then all the D_j's are identical up to reflection and H can be simply chosen as D_1.

3 Complexity Issues

Next, we shall consider five simple special cases of this problem that will shed some light on the complexity of this problem. The first case corresponds to the situation where there is no sizing error; however, there may be false cuts and missing cuts (due to partial digestion). On the other hand, we make the strong assumption that a precise lower bound on the partial digestion rate is available. The second case corresponds to the situation where there is no false cut, but there is some sizing error. However, we assume that rough location of the cut sites may be known in advance. In either case, we assume that for some fraction of the single molecule restriction maps the correct orientation is not known. The third case corresponds to the situation where an end fragment (either left or right) is missing, but it may be uncertain which end the fragment is missing from. (In this case, the orientation of the molecule may be assumed to be known.) The last two problems model the situations where we may have spurious data or data from k (≥ 2) distinct populations. In all situations, the notion of "consistency" can be defined in the most natural manner. We make these notions more precise.

3.0.1 Problem 1 (Unknown Orientation)

Given a set of ordered vectors with rational entries in the open interval $(0,1)$:

$$D_1, D_2, \ldots, D_l, D_{l+1}, \ldots, D_m,$$

a rational number $p_c \in (0,1)$ and an integer N.

An admissible alignment of the data can be represented as

$$D_1', D_2', \ldots, D_l', D_{l+1}', \ldots, D_m',$$

where $D_j' \in \{D_j, D_j^R\}$ $(1 \leq j \leq l)$ and $D_j' = D_j$ $(j > l)$. For any such alignment (A_k) and a rational number $h_i \in [0,1]$, define an indicator variable m_{ijk} to be 1, if $h_i \in D_j'$ and 0, otherwise. Now, define a characteristic function $\chi_k : [0,1] \to \{0,1\}$, as $\chi_k(h_i) = 1$, iff $\sum_j m_{ijk} > p_c m$.

Determine: If there is an admissible alignment A_k such that

$$|\{h \in [0,1] | \chi_k(h) = 1\}| \geq N.$$

This decision problem plays a crucial role in formulating a binary search algorithm to solve the following optimization problem: assuming that the data support an N-cut solution, find a final restriction map H^* with no fewer than N cuts where each cut is supported independently by $p_c^* m$ matches or more, where p_c^* attains the maximal possible value. Note that the two parameters N and p_c are interrelated and cannot be optimized independently. In practice, however, the values of N can be characterized quite accurately by means of its distribution from some known prior information. The dual problem of estimating p_c however do not work as well, since the chemical processes that govern the digestion process (particularly on a stretched molecule on a surface) is difficult to model with any significant accuracy. Another approach would be to formulate the problem in terms of a weight function that are monotonically non-decreasing in both p_c and N. Under some reasonable assumptions, it can be shown that the worst-case complexity of the problem remains unchanged.

We shall show that problem 1 is NP-complete for size m. The problem is clearly NP-computable, since if one can guess a correct admissible alignment, then it is easy to check in polynomial time if there are no fewer than N restriction sites.

The NP-hardness of the problem can be easily shown with a simple transformation from 3-SAT. Consider an instance of a 3-SAT problem with l variables, x_1, x_2, ..., x_l and n clauses, C_1, C_2, ..., C_n (we may take $n \geq l$). Without loss of generality, we assume that no clause contains a variable x_j and its negation $\bar{x}_j$, since such a clause always assumes true value, independent of x_j's truth value. With each clause, C_i associate a location $f_i = i/2(n+1) \in (0, 1/2)$ and $f_i^R = 1 - f_i = (2n - i + 2)/2(n+1) \in (1/2, 1)$. The problem is NP-hard in size l. For each instance of the 3-SAT problem we create a data set D_1, ..., D_l, D_{l+1}, ..., D_m, with $m = 2l - 1$ as follows: Each D_j will have cuts only at f_i's and f_i^R's. We will have total $m = 2l - 1$ data items, where the first l data items may need to be reoriented, but the last $l - 1$ items are in correct orientation:

$$D_{l+1} = \cdots = D_m = (f_1, f_2, \ldots, f_n)$$

The first l, D_j's are determined as follows. $f_i \in D_j$, iff $x_j \in C_i$ and $f_i^R \in D_j$, iff $\bar{x}_j \in C_i$. Of course, we choose $N \equiv n$ and $p_c = 1/2$. If the CNF has a satisfying assignment, then choose an admissible alignment, in which $D_j' = D_j$, if $x_j =$ True and $D_j' = D_j^R$, if $x_j =$ False (for $1 \leq j \leq l$). The last $l - 1$ data items are left untouched. Clearly, for every f_i $(1 \leq i \leq n)$ there are $l - 1$ "matches" from the data items D_{l+1}, ..., D_m and at least one more from D_1', ..., D_l' (since each clause must have been satisfied). Thus, for each i $(1 \leq i \leq n)$, $\sum_j m_{ijk} \geq l > p_c m = (2l-1)/2$ and $\chi_k(f_i) = 1$ and for all $h \neq f_i$, $\chi_k(h) = 0$.

$$\{h \in [0,1] | \chi_k(h) = 1\} \ = \ \{f_1, f_2, \ldots, f_n\}$$

and

$$|\{h \in [0,1] | \chi_k(h) = 1\}| \ = \ n.$$

Conversely, it is rather easy to see that if the CNF has no satisfying assignment, then for every admissible alignment k there exists an i $(1 \leq i \leq n)$ with $\sum_j m_{ijk} = l-1 < p_c m = (2l-1)/2$ and hence

$$|\{h \in [0,1] | \chi_k(h) = 1\}| < n.$$

3.0.2 Problem 2 (Sizing Errors)

Given a set of ordered vectors with rational entries in the open interval $(0,1)$:

$$D_1, D_2, \ldots, D_l, D_{l+1}, \ldots, D_m,$$

an approximate solution

$$\tilde{H} = (\tilde{h}_1, \ldots, \tilde{h}_N),$$

an approximation factor ϵ and a variance upper bound σ^2.

An admissible alignment of the data can be represented as

$$D'_1, D'_2, \ldots, D'_l, D'_{l+1}, \ldots, D'_m,$$

where $D'_j \in \{D_j, D_j^R\}$ $(1 \leq j \leq l)$ and $D'_j = D_j$ $(j > l)$, as before. For any such alignment (A_k) and an approximate cut site $\tilde{h}_i$ define a set

$$S_{ijk} = \{s \in D'_j | \ \ |s - \tilde{h}_i| \leq \epsilon\}$$

and

$$S_{ik} = \biguplus_j S_{ijk}$$

Define $h_i = \mathrm{mean}(S_{ik})$ and $\sigma_i^2 = \mathrm{var}(S_{ik})$.

Determine: If there is an admissible alignment A_k such that

$$\forall_i \ \ \sigma_i^2 \leq \sigma^2.$$

We shall show that problem 2 is NP-complete in the size m. The problem is clearly NP-computable, since if one can guess a correct admissible alignment, then it is easy to check in polynomial time if the variance bound can be met.

The NP-hardness of the problem can be easily shown with a simple transformation from NOT-ALL-EQUAL 3-SAT. Consider an instance of NOT-ALL-EQUAL 3-SAT problem with l variables, x_1, x_2, …, x_l and n clauses, C_1, C_2, …, C_n (with $n \geq l$). As before, we assume that no clause contains a variable x_j and its negation $\bar{x}_j$, since such a clause has always one true literal and one false literal, independent of x_j's truth value. This problem is NP-hard in size l. Consider a given instance of NOT-ALL-EQUAL 3-SAT problem; with each clause, C_i associate locations $f_i = i/2(n+1)$, $g_i = f_i - 1/5(n+1)$, f_i^R and g_i^R. Each D_j will have cuts only at f_i's, g_i's, f_i^R's and g_i^R's. We will have total $m = l + 3$ data items, where the first l data items may need to be reoriented, but the last 3 items are in correct orientation:

$$D_{l+1} = \cdots = D_m = (f_1, \ldots, f_n, f_n^R, \ldots, f_1^R)$$

The first l, D_j's are determined as follows. $g_i \in D_j$, iff $x_j \in C_i$ and $g_i^R \in D_j$, iff $\bar{x}_j \in C_i$. We set

$$\tilde{H} = (f_1, \ldots, f_n, f_n^R, \ldots, f_1^R)$$

and $\epsilon = 1/5(n+1)$ and $\sigma^2 = 6/625(n+1)^2$.

If the CNF has a satisfying assignment such that each clause has at least one true literal and at least one false literal, then choose an admissible alignment, in which $D'_j = D_j$, if $x_j =$ True and $D'_j = D_j^R$, if $x_j =$ False (for $1 \le j \le l$). The last 3 data items are left untouched. Clearly, for every $\tilde{h}_i$ $(1 \le i \le n)$ there are 3 "matches" from the data items D_{l+1}, ..., D_m with a value f_i and exactly one or two additional matches from D'_1, ..., D'_l with a value g_i. Thus the variance of S_{ik} is $\sigma_i^2 \le \mathrm{var}(0, 0, 0, \epsilon, \epsilon) = (2/5)(3/5)(1/5(n+1))^2 = 6/625(n+1)^2$. A similar argument shows that variances for $n+1 \le i \le 2n$ are also similarly bounded.

If on the other hand, for every truth assignment there exists a clause C_i that must have its literals all True or all False, we see that either $\sigma_i^2 = \mathrm{var}(0, 0, 0, \epsilon, \epsilon, \epsilon) = (1/2)(1/2)(1/5(n+1))^2 = 1/100(n+1)^2$ or $\sigma_{2n-i}^2 = 1/100(n+1)^2$. In either case, either $\sigma_i^2 > \sigma^2$ or $\sigma_{2n-i}^2 > \sigma^2$. The following observation is sufficient to derive these values: if C_i has all its literals true, then

$$S_{ik} = \{g_i, g_i, g_i, f_i, f_i, f_i\} \quad \text{and} \quad S_{2n-i,k} = \{f_i^R, f_i^R, f_i^R\}.$$

If C_i has all its literals false, then

$$S_{ik} = \{f_i, f_i, f_i\} \quad \text{and} \quad S_{2n-i,k} = \{g_i^R, g_i^R, g_i^R, f_i^R, f_i^R, f_i^R\}.$$

3.0.3 Problem 3 (Missing Fragments)

Given a set of ordered vectors with rational entries in the open interval $(0, 1)$:

$$D_1, D_2, \ldots, D_l, D_{l+1}, \ldots, D_m,$$

a rational number $p_c \in (0, 1)$ and an integer N.

An admissible alignment of the data can be represented as

$$D'_1, D'_2, \ldots, D'_l, D'_{l+1}, \ldots, D'_m,$$

where $D'_j \in \{D_j, D_j + c_j\}$ $(0 < c_j < 1 - s_{M_j,j}$, $1 \le j \le l)$ and $D'_j = D_j$ $(j > l)$. Observe that in this case, we have assumed that the orientations of the molecules are known, but the first l molecules may have *missing* fragments on either end. The possibility of a missing end fragment is not a problem for small-sized clones (e.g., cosmids), but do pose serious difficulties for larger clones (i.e., BACs).

For any such alignment (A_k) and a rational number $h_i \in [0, 1]$, define an indicator variable m_{ijk} to be 1, if $h_i \in D'_j$ and 0, otherwise. Now, define a characteristic function $\chi_k : [0, 1] \to \{0, 1\}$, as $\chi_k(h_i) = 1$, iff $\sum_j m_{ijk} > p_c m$.

Determine: If there is an admissible alignment A_k such that

$$|\{h \in [0, 1] | \chi_k(h) = 1\}| \ge N.$$

We shall show that problem 3 is also NP-complete in size m. The problem is clearly NP-computable, since if one can guess a correct admissible alignment, then it is easy to check in polynomial time if there are no fewer than N restriction sites.

The NP-hardness of the problem can be easily shown with a simple transformation from 3-SAT. Consider an instance of a 3-SAT problem with l variables, x_1, x_2, …, x_l and n clauses, C_1, C_2, …, C_n (with $n \geq l$). Without loss of generality, we assume that no clause contains a variable x_j and its negation $\bar{x}_j$, since such a clause always assumes true value, independent of x_j's truth value. For a given instance of 3-SAT problem, we proceed as follows: With each clause, C_i associate a location $f_i = i/(n+1) \in (0,1)$ and $g_i = (i - 1/2)/(n+1) \in (0,1)$. Each D_j will have cuts only at f_i's and g_i's. We will have total $m = 2l - 1$ data items, where the first l data items may need to be translated, but the last $l - 1$ items are in correct orientation:

$$D_{l+1} = \cdots = D_m = (f_1, f_2, \ldots, f_n)$$

The first l, D_j's are determined as follows. $f_i \in D_j$, iff $x_j \in C_i$ and $g_i \in D_j$, iff $\bar{x}_j \in C_i$. Of course $N \equiv n$ and $p_c = 1/2$. If the CNF has a satisfying assignment, then choose an admissible alignment, in which $D'_j = D_j$, if $x_j =$ True and $D'_j = D_j + 1/2(n+1)$, if $x_j =$ False (for $1 \leq j \leq l$). The last $l - 1$ data items are left untouched. Clearly, for every f_i $(1 \leq i \leq n)$ there are $l - 1$ "matches" from the data items D_{l+1}, …, D_m and at least one more from D'_1, …, D'_l (since each clause must have been satisfied). Thus, for each i $(1 \leq i \leq n)$, $\sum_j m_{ijk} \geq l > p_c m$ and $\chi_k(f_i) = 1$ and for all $h \neq f_i$, $\chi_k(h) = 0$.

$$\{h \in [0,1] | \chi_k(h) = 1\} \; = \; \{f_1, f_2, \ldots, f_n\}$$

and

$$|\{h \in [0,1] | \chi_k(h) = 1\}| \; = \; n.$$

Conversely, it is rather easy to see that if the CNF has no satisfying assignment, then for every admissible alignment k

$$|\{h \in [0,1] | \chi_k(h) = 1\}| < n.$$

3.0.4 Problem 4 (Spurious Data)

Next, we model the effect of the fact that some of the data items may be invalid ("bad"). Thus these data items can be assumed to have no relation to the restriction map being computed and hence should be discarded before the final restriction map is computed. However, we may assume that each molecule is given with the correct orientation and that the fragments are correctly sized. The number of bad molecules is also assumed to be known and is exactly p_b fraction of the total number of molecules. The bound on the digestion rate is denoted as before by p_c.

Given a set of ordered vectors with rational entries in the open interval $(0,1)$:

$$\mathcal{D} = \{D_1, D_2, \ldots, D_m\},$$

two rational numbers p_c, $p_b \in (0,1)$ and an integer N.

The set $\mathcal{D}$ may then be partitioned into two subsets $\mathcal{D}_k$ (the "good" molecules) and $\mathcal{D}_k^c$ (the "bad" molecules). The final restriction map is then computed using only the $\mathcal{D}_k$ with respect to the alignment given (implicit in the way $\mathcal{D}$ is presented).

By definition, $|\mathcal{D}_k^c| = p_b|\mathcal{D}|$. For any such subset $\mathcal{D}_k$ and a rational number $h_i \in [0,1]$, define an indicator variable m_{ijk} to be 1, if $h_i \in D_j$ and $D_j \in \mathcal{D}_k$ and 0, otherwise. Now, define a characteristic function $\chi_k : [0,1] \to \{0,1\}$, as $\chi_k(h_i) = 1$, iff $\sum_j m_{ijk} > p_c|\mathcal{D}_k|$.

Determine: If there is a subset of "good" molecules $\mathcal{D}_k$ such that

$$|\{h \in [0,1] | \chi_k(h) = 1\}| \geq N.$$

We shall show that problem 4 is NP-complete in size m. The problem is clearly NP-computable, since if one can guess a correct subset of good molecules, then it is easy to check in polynomial time if there are no fewer than N restriction sites.

The NP-hardness of the problem can be shown by a transformation from 3-SAT. Consider an instance of a 3-SAT problem with l variables x_1, x_2, ..., x_l and n clauses, C_1, C_2, ..., C_n (with $n \geq l$). As before, without loss of generality, we assume that no clause contains a variable x_j and its negation $\bar{x}_j$, since such a clause always evaluates to true, independent of x_j's truth value. For a given instance of 3-SAT problem, we proceed as follows: With each variable, x_j associate a location $g_j = j/2(l+1) \in (0, 1/2)$; with each clause, C_i associate a location $f_i = 1/2 + i/4(n+1) \in (1/2, 3/4)$ and finally, a location $e = 7/8$. Each D_j will have cuts only at f_i's, g_j's and e. We will have total $m = 4l - 2$ data items, all of them in correct alignment. The first $2l$, D_j's ($1 \leq j \leq 2l$) are determined as follows: There are two data items D_{2j-1} and D_{2j} for each variable x_j. $g_j \in D_{2j-1}$ and $g_j \in D_{2j}$. $f_i \in D_{2j-1}$, iff $x_j \in C_i$ and $f_i \in D_{2j}$, iff $\bar{x}_j \in C_i$. The next $l-1$, D_j's ($2l < j \leq 3l-1$) are given as

$$D_{2l+1} = \cdots = D_{3l-1} = (g_1, \ldots, g_l, f_1, \ldots, f_n).$$

The last $l-1$, D_j's ($3l - 1 < j < 4l - 2 = m$) are given as

$$D_{3l} = \cdots = D_m = (e).$$

Finally, $N \equiv n + l$ and $p_c = p_b = 1/2$

If the CNF has a satisfying assignment, then choose a partition of $\mathcal{D}$ as follows: For $1 \leq j \leq 2l$, $D_{2j-1} \in \mathcal{D}_k$, if $x_j =$ True and $D_{2j} \in \mathcal{D}_k$, if $x_j =$ False. For $2l < j \leq 3l-1$, $D_j \in \mathcal{D}_k$. The rest of the molecules are in $\mathcal{D}_k^c$. Clearly the number of bad molecules is $l + (l-1) = 2l - 1 = p_b m$. Also $\mathcal{D}_k$ has exactly $2l-1$ elements. Since either D_{2j-1} or D_{2j} is in $\mathcal{D}_k$, $\chi_k(g_j) = 1$ (There are $l \geq p_c(2l-1)$ cuts at each location g_j in $\mathcal{D}_k$). Since every clause is satisfied individually, $\chi_k(f_i) = 1$ (There are at least $l \geq p_c(2l-1)$ cuts at each location f_i in $\mathcal{D}_k$). Clearly $\chi_k(e) = 0$.

$$\{h \in [0,1] | \chi_k(h) = 1\} \ = \ \{g_1, \ldots, g_l, f_1, \ldots, f_n\},$$

and

$$|\{h \in [0,1] | \chi_k(h) = 1\}| \ = \ n + l.$$

Conversely, if $\mathcal{D}$ can be partitioned so that $\mathcal{D}_k$ (the "good" molecules) satisfies all the constraints, then we make the following observations about $\mathcal{D}_k$: (1) For $3l \leq j \leq 4l-2$,

$D_j \notin \mathcal{D}_k$. (2) For $2l+1 \leq j \leq 3l-1$, $D_j \in \mathcal{D}_k$. (3) For $1 \leq j \leq l$, exactly one of D_{2j-1} and D_{2j} must be in $\mathcal{D}_k$. Otherwise, either there will not be enough cuts at g_j in $\mathcal{D}_k$ or the subset $\mathcal{D}_k^c$ will not have half (p_b fraction) of all the data elements. From this it follows that at each f_i location there will be at least l cuts in $\mathcal{D}_k$ and the corresponding clause will evaluate to true leading to a truth assignment in which x_j is true (respectively, false) if $D_{2j-1} \in \mathcal{D}_k$ (respectively, $D_{2j} \in \mathcal{D}_k$).

3.0.5 Problem 5 (k-Populations)

Finally, we model a closely related problem, in which each of the data items is assumed to correspond to exactly one of k different restriction maps. Thus one assumes that the optical map data was derived by taking a mixture of k different clones and then cleaving them by a restriction enzyme. Thus computationally our job is to partition the data items into k "equivalent classes" and then compute k restriction maps corresponding to each of the k subsets. We show that this problem is NP-complete even when $k = 2$, all the orientations are known and there is no sizing error. The construction is very similar to the one for problem 4, but we present it for the sake of completeness.

Given a set of ordered vectors with rational entries in the open interval $(0, 1)$:

$$\mathcal{D} = \{D_1, D_2, \ldots, D_m\},$$

two rational numbers p_1, $p_2 \in (0, 1)$ and two integers N_1 and N_2.

The set $\mathcal{D}$ may then be partitioned into two subsets $\mathcal{D}_k$ (the type 1 molecules) and $\mathcal{D}_k^c$ (the type 2 molecules). The final type 1 (respectively, type 2) restriction map is then computed using only the $\mathcal{D}_k$ (respectively, $\mathcal{D}_k^c$) with respect to the alignment given. For the sake of simplicity, we shall assume that $|\mathcal{D}_k| = |\mathcal{D}_k^c|$.

For any such subset $\mathcal{D}_k$ and a rational number $h_i \in [0, 1]$, define an indicator variable m_{ijk} to be 1, if $h_i \in D_j$ and $D_j \in \mathcal{D}_k$ and 0, otherwise. Now, define a characteristic function $\chi_k : [0, 1] \to \{0, 1\}$, as $\chi_k(h_i) = 1$, iff $\sum_j m_{ijk} > p_1 |\mathcal{D}_k|$. Similarly, define χ_k^c, with $\mathcal{D}_k^c$ and p_2 taking the roles of $\mathcal{D}_k$ and p_1, respectively.

Determine: If there is an equi-partition of $\mathcal{D} = \mathcal{D}_k \cup \mathcal{D}_k^c$ such that

$$\begin{aligned} |\{h \in [0,1] | \chi_k(h) = 1\}| &\geq N_1, \quad \text{and} \\ |\{h \in [0,1] | \chi_k^c(h) = 1\}| &\geq N_2. \end{aligned}$$

Problem 5 is NP-complete in size m. The problem is clearly NP-computable, since if one can guess a correct partition, then it is easy to check in polynomial time if the two maps have no fewer than N_1 and N_2 restriction sites, respectively.

The NP-hardness of the problem can be shown by a transformation from 3-SAT. Consider an instance of a 3-SAT problem with l variables x_1, x_2, ..., x_l and n clauses, C_1, C_2, ..., C_n (with $n \geq l$). As before, without loss of generality, we assume that no clause contains a variable x_j and its negation $\bar{x}_j$, since such a clause always evaluates to true, independent of x_j's truth value. For a given instance of 3-SAT problem, we proceed as follows: With each variable, x_j associate a location $g_j = j/2(l+1) \in (0, 1/2)$; with each clause, C_i associate two locations $f_i = 1/2 + i/4(n+1) \in (1/2, 3/4)$ and $f_i' = 3/4 + i/4(n+1) \in (3/4, 1)$. Each

D_j will have cuts only at f_i's, f_i''s and g_j's. We will have total $m = 4l - 2$ data items, all of them in correct alignment. The first $2l$, D_j's $(1 \le j \le 2l)$ are determined as follows: There are two data items D_{2j-1} and D_{2j} for each variable x_j. $g_j \in D_{2j-1}$ and $g_j \in D_{2j}$. $f_i \in D_{2j-1}$ and $f_i' \in D_{2j}$, iff $x_j \in C_i$ and $f_i' \in D_{2j-1}$ and $f_i \in D_{2j}$, iff $\bar{x}_j \in C_i$. The next $l-1$, D_j's $(2l < j \le 3l-1)$ are given as

$$D_{2l+1} = \cdots = D_{3l-1} = (g_1, \ldots, g_l, f_1, \ldots, f_n).$$

The last $l-1$, D_j's $(3l - 1 < j < 4l - 2 = m)$ are given as

$$D_{3l} = \cdots = D_m = (g_1, \ldots, g_l, f_1', \ldots, f_n').$$

Finally, $N_1 = N_2 \equiv n + l$ and $p_1 = p_2 = 1/2$

If the CNF has a satisfying assignment, then choose a partition of $\mathcal{D}$ as follows: For $1 \le j \le 2l$, $D_{2j-1} \in \mathcal{D}_k$, if $x_j =$ True and $D_{2j} \in \mathcal{D}_k$, if $x_j =$ False. For $2l < j \le 3l-1$, $D_j \in \mathcal{D}_k$. The rest of the molecules are in $\mathcal{D}_k^c$. Both $\mathcal{D}_k$ and $\mathcal{D}_k^c$ have exactly $2l-1$ elements. Since either D_{2j-1} or D_{2j} is in $\mathcal{D}_k$ (respectively $\mathcal{D}_k^c$), $\chi_k(g_j) = 1$ (respectively, $\chi_k^c(g_j) = 1$). Since every clause is satisfied individually, $\chi_k(f_i) = 1$ and $\chi_k^c(f_i') = 1$.

$$\{h \in [0,1] | \chi_k(h) = 1\} \; = \; \{g_1, \ldots, g_l, f_1, \ldots, f_n\},$$

and

$$\{h \in [0,1] | \chi_k^c(h) = 1\} \; = \; \{g_1, \ldots, g_l, f_1', \ldots, f_n'\}.$$

Thus

$$|\{h \in [0,1] | \chi_k(h) = 1\}| \; = \; |\{h \in [0,1] | \chi_k^c(h) = 1\}| \; = \; n + l.$$

Conversely, if $\mathcal{D}$ can be partitioned so that the two subsets satisfy all the constraints, then we claim that there always exist a partition satisfying the following additional properties: (1) For $2l+1 \le j \le 3l-1$, $D_j \in \mathcal{D}_k$. (2) For $3l \le j \le 4l-2$, $D_j \in \mathcal{D}_k^c$. (3) For $1 \le j \le l$, exactly one of D_{2j-1} and D_{2j} must be in $\mathcal{D}_k$. Otherwise, there will not be enough cuts at g_j in $\mathcal{D}_k$ or its complement. From this it follows that at each f_i location there will be at least l cuts in $\mathcal{D}_k$ and the corresponding clause will evaluate to true leading to a truth assignment in which x_j is true (respectively, false) if $D_{2j-1} \in \mathcal{D}_k$ (respectively, $D_{2j-1} \in \mathcal{D}_k^c$).

4 Efficient Probabilistic Algorithm

In spite of the pessimistic results of the previous section, it is not hard to see that the problem admits efficient algorithms once the structure in the input is exploited. For instance, if the digestion rate is quite high, then by looking at the distribution of the cuts a good guess can be made about the number of cuts and then only the data set with large numbers of cuts can be combined to create the final map [Ree97]. Other approaches have used formulations in which one optimizes a cost function and provides heuristics (as the exact optimization problems are often infeasible). In one approach[2], the optimization problem corresponds to finding weighted cliques; and in another [MP96], the formulation corresponds to a 0-1

[2]An earlier algorithm due to the first author.

quadratic programming problem. However, these heuristics have only worked on limited sets of data and their effectivity (or approximability) remains unproven.

Here, we present a probabilistic algorithm based on a Bayesian approach. Our approach is to use a carefully constructed prior model of the cuts to infer the best hypothetical model by using Bayes' formula[DLR77, GM93]. The solution requires searching over a high-dimensional hypothesis space and is complicated by the fact that the underlying distributions are multimodal. We show how the search over this space can be accomplished without sacrificing efficiency. The algorithm has been implemented and extensively tested over automatically generated data for more than a year with good results (see section 5). Furthermore, one can speed up the algorithm quite easily by suitably constraining various parameters in the implementation (but at the loss of accuracy or an increased probability of missing the correct map).

The main ingredients of this Bayesian scheme are the following:

- A Model or Hypothesis $\mathcal{H}$, of the map of restriction sites.
- A Prior distribution of the data (SMRM vectors)

  $$\mathbb{P}\mathrm{r}[D_j|\mathcal{H}],$$

  Assume pair-wise conditional independence of the data (SMRM) vectors D_j

  $$\mathbb{P}\mathrm{r}[D_j|D_{j_1},\ldots,D_{j_m},\mathcal{H}] = \mathbb{P}\mathrm{r}[D_j|\mathcal{H}],$$

  Thus, the prior of the entire data set of SMRM vectors becomes

  $$\mathbb{P}\mathrm{r}[\mathcal{D}|\mathcal{H}] = \prod_j^m \mathbb{P}\mathrm{r}[D_j|\mathcal{H}],$$

  where index j ranges over the data set.
- The Posterior distributions via Bayes' rule

  $$\mathbb{P}\mathrm{r}[\mathcal{H}|\mathcal{D}] = \frac{\mathbb{P}\mathrm{r}[\mathcal{D}|\mathcal{H}]\ \mathbb{P}\mathrm{r}[\mathcal{H}]}{\mathbb{P}\mathrm{r}[\mathcal{D}]}$$

Using this formulation, we search over the space of all hypotheses to find the most "plausible" hypothesis $\mathcal{H}^*$ that maximizes the posterior probability. Here $\mathbb{P}\mathrm{r}[\mathcal{H}]$ is the prior unconditional distribution of hypothesis $\mathcal{H}$, and $\mathbb{P}\mathrm{r}[\mathcal{D}]$ is the unconditional distribution of the data.

The hypotheses $\mathcal{H}$ will be modeled by a small number of parameters $\Phi(\mathcal{H})$ (e.g., number of cuts, distributions of the cuts, distributions of the false cuts, etc.). We have prior models for only a few of these parameters (number of cuts), and the other parameters are implicitly assumed to be equi-probable. Thus the model of $\mathbb{P}\mathrm{r}[\mathcal{H}]$ is rather simplistic. The unconditional distributions for the data $\mathbb{P}\mathrm{r}[\mathcal{D}]$ does not have to be computed at all since it does not effect the choice of $\mathcal{H}^*$. In contrast, we use a very detailed model for the conditional distribution for the data given the chosen parameter values for the hypothesis. One can write the above expression as

$$\log(\mathbb{P}\mathrm{r}[\Phi(\mathcal{H})|\mathcal{D}]) = \mathcal{L} + \text{Penalty} + \text{Bias},$$

where $\mathcal{L} \equiv \sum_j \log(\mathbb{P}\mathrm{r}[D_j|\Phi(\mathcal{H})])$ is the *likelihood function*, Penalty $= \log \mathbb{P}\mathrm{r}[\hat{\Phi}(\mathcal{H})]$ and Bias $= -\log(\mathbb{P}\mathrm{r}[\mathcal{D}])$ = a constant. In these equations $\Phi(\mathcal{H})$ corresponds to the parameter set describing the hypothesis and $\hat{\Phi}(\mathcal{H}) \subseteq \Phi(\mathcal{H})$ a subset of parameters that have a nontrivial prior model. In the following, we shall often write $\mathcal{H}$ for $\Phi(\mathcal{H})$, when the context creates no ambiguity.

Also, note that the bias term has no effect as it is a constant (independent of the hypothesis), and the penalty term has any discernible effect only when the data set is small. Thus, our focus is often on the term $\mathcal{L}$ which dominates all other terms in the right hand side.

As we will see the posterior density, $\mathbb{P}\mathrm{r}[\mathcal{H}|\mathcal{D}]$ is multimodal and the prior $\mathbb{P}\mathrm{r}[D_j|\mathcal{H}]$ does not admit a closed form evaluation (as it is dependent on the orientation and alignment with $\mathcal{H}$). Thus, we need to rely on iterative sampling techniques.

Thus the algorithm has two parts: we take a sample hypothesis and locally search for the most plausible hypothesis in its neighborhood using gradient search techniques; we use a global search to generate a set of sample hypotheses and filter out all but the ones that are likely to be near plausible hypotheses. We present the algorithmic descriptions of the local and global searches in that order.

Note that our approach based on the Bayesian scheme enjoys many advantages:

- One obtains the best possible estimate of map given the data, subject only to the comprehensiveness of the model $\Phi(\mathcal{H})$ used.
- For a comprehensive model $\mathcal{H}$, estimates of $\Phi(\mathcal{H})$ are unbiased and errors converge asymptotically to zero as data size increases.
- Additional sources of error can be modeled simply by adding parameters to $\Phi(\mathcal{H})$.
- Estimates of the errors in the result can be computed in a straightforward manner.
- The algorithm provides an easy way to compute a quality measure.

However, the approach also has many shortcomings:

- It is computationally expensive to compute $\mathbb{P}\mathrm{r}[\Phi(\mathcal{H})|\mathcal{D}]$.
- The search for best $\Phi(\mathcal{H})$ is often expensive since posterior distributions are typically multimodal.
- Typically sampling on the parameter space is the only option. However this sampling is not exhaustive, and hence the best $\Phi(\mathcal{H})$ may not always be found.
- Good prior $\mathbb{P}\mathrm{r}[\Phi(\mathcal{H})]$ may not be available, requiring more data for the same "quality of goodness" and may introduce bias.
- The quality measure and parameter accuracy estimates may be incorrect if the data sample size is small.

4.1 Maps by Bayesian Inference

In order to accurately model the prior observation distribution $\mathbb{P}\mathrm{r}[\mathcal{D}|\mathcal{H}]$, we need to consider following categories of errors in image data: 1) Misidentification of spurious materials in the image as DNA, 2) Identifying multiple DNA molecules as one, 3) Identifying partial DNA molecules as complete, 4) Errors in estimating sizes of DNA fragments, 5) Incomplete digestion of DNA, 6) Cuts visible at locations other than digest sites, and 7) Orientation of DNA molecule is not always known.

Our observation probability distribution $\mathbb{P}\mathrm{r}[\mathcal{D}|\mathcal{H}]$ is modeled as following:

- A molecule on the surface can be read from left to right or right to left. The uncertainty in orientation is modeled as Bernoulli processes, with the probability for each orientation being equal.

- The restrictions sites on the molecule are determined by a distribution induced by the underlying distribution of the four bases in the DNA. For example, we shall assume that the probability that a particular base (say, A) appears at a location i is independent of the other bases, though the probabilities are not necessarily identical.

- The false cuts appear on the molecule as a Poisson process. This is a consequence of the simplifying assumption that over a small region Δh on the molecule, the $\mathbb{P}\mathrm{r}[\#$ False cuts $= 1$ over $\Delta h] = \lambda_f \Delta h$ and the $\mathbb{P}\mathrm{r}[\#$ False cuts ≥ 2 over $\Delta h] = o(\Delta h)$.

- The fragment size (the size of the molecule between two cuts) is estimated with some loss of accuracy (dependent on the stretching of the molecule, fluorochrome attachments and the image processing algorithm). The measured size is assumed to be distributed as a Gaussian.

Following notation will be used to describe the parameters of the independent processes responsible for the statistical structure of the data. Unless otherwise specified, the indices i, j and k are to have the following interpretation:

- The index i ranges from 1 to N and refers to cuts in the hypothesis.

- The index j ranges from 1 to M and refers to data items (i.e., molecules).

- The index k ranges from 1 to K and refers to a specific alignment of cuts in the hypothesis vs. data.

Now the main parameters of our Bayesian model are as follows:

- p_{c_i} = Probability that the ith sequence specific restriction site in the molecule will be visible as a cut.

- σ_i = Standard deviation of the observed position of the ith cut when present and depends on the accuracy with which a fragment can be sized.

- λ_f = Expected number of false-cuts per molecule observed. Since all sizes will be normalized by the molecule size, this will also be the false-cuts per unit length.

- p_b = Probability that the data is invalid ("bad"). In this case, the data item is assumed to have no relation to the hypothesis being tested, and could be an unrelated piece of DNA or a partial molecule with a significant fraction of the DNA missing. The cut-sites (all false) on this data item are assumed to have been generated by a Poisson process with the expected number of cuts = λ_n.

  Note that the regular DNA model reduces to the "bad" DNA model for the degenerate situation when $p_{c_i} \to 0$ and $\lambda_f \to \lambda_n$. As a result, "bad" DNA molecules cannot be disambiguated from regular DNA molecules if $p_{c_i} \approx 0$. In practice, $p_{c_i} > 0$ and $\lambda_n > \lambda_f$, and the degenerate case almost never occurs. Here the "bad" molecules are recognized by having a disproportionately large number of false cuts.

- λ_n = Expected number of cuts per "bad" molecule.

Recall that by Bayes' rule

$$\mathbb{P}\mathrm{r}[\mathcal{H}|\mathcal{D}] = \frac{\mathbb{P}\mathrm{r}[\mathcal{D}|\mathcal{H}]\ \mathbb{P}\mathrm{r}(\mathcal{H})}{\mathbb{P}\mathrm{r}[\mathcal{D}]}$$

Assuming that the prior $\mathbb{P}\mathrm{r}[\mathcal{H}]$ distribution is given (as in the following subsection) in terms of just the number of restriction sites, based on the standard Poisson distribution, we wish to find the "most plausible" hypothesis $\mathcal{H}$ by maximizing $\mathbb{P}\mathrm{r}[\mathcal{D}|\mathcal{H}]$.

In our case, $\mathcal{H}$ is simply the final map (a sequence of restriction sites, h_1, h_2, …, h_N) augmented by the auxiliary parameters such as p_{c_i}, σ_i, λ_f, etc. When we compare a data item D_j with respect to this hypothesis, we need to consider every possible way that D_j could have been generated by $\mathcal{H}$. In particular we need to consider every possible alignment, where the k^{th} alignment, A_{jk}, corresponds to a choice of the orientation for D_j as well as identifying a cut on D_j with a true restriction site on $\mathcal{H}$ or labeling the cut as a false cut. By $D_j^{(A_{jk})}$ [also abbreviated as $D_j^{(k)}$], we shall denote the "interpretation of the jth data item with respect to the alignment A_{jk}." Each alignment describes an independent process by which D_j could have been generated from $\mathcal{H}$, and therefore the total probability density of D_j is the sum of the probability density of all these alignments, plus the remaining possible derivations (invalid data).

As a consequence of the pairwise independence and the preceding discussion, we have the following:

$$\begin{aligned}
\mathbb{P}\mathrm{r}[\mathcal{D}|\mathcal{H}] &= \prod_j^M \mathbb{P}\mathrm{r}[D_j|\mathcal{H}], \\
&\quad \text{where index } j \text{ ranges over the data set.} \\
\mathbb{P}\mathrm{r}_j \equiv \mathbb{P}\mathrm{r}[D_j|\mathcal{H}] &= \frac{1}{2}\sum_k \mathbb{P}\mathrm{r}[D_j^{(k)}|\mathcal{H},\mathrm{good}]\ \mathbb{P}\mathrm{r}[\mathrm{good}] + \frac{1}{2}\sum_k \mathbb{P}\mathrm{r}[D_j^{(k)}|\mathcal{H},\mathrm{bad}]\ \mathbb{P}\mathrm{r}[\mathrm{bad}] \\
&\quad \text{where index } k \text{ ranges over the set of alignments.}
\end{aligned}$$

In the preceding equation, $\mathbb{P}\mathrm{r}[D_j^{(k)}|\mathcal{H},\mathrm{good}]$ (abbreviated, $\mathbb{P}\mathrm{r}_{jk}$) is the probability density of model D_j being derived from model $\mathcal{H}$ and corresponding to a particular alignment of cuts (denoted, A_{jk}). The set of alignments include alignments for both orientations, hence each alignment has a prior probability of 1/2. If D_j is bad, our model corresponds to $\mathcal{H}$ with $p_{c_i} \to 0$ and $\lambda_f \to \lambda_n$. We shall often omit the qualifier "good" for the hypothesis $\mathcal{H}$, when it is clear from the context.

Thus, in the example shown in Figure 2, for a given hypothesis $\mathcal{H}$, the conditional probability density that the j^{th} data item D_j with respect to alignment A_{jk} (i.e., $D_j^{(k)}$) could have occurred is given by the following formula:

$$\mathbb{P}\mathrm{r}_{jk} = p_{c_1} \frac{e^{-(s_1-h_1)^2/2\sigma_1^2}}{\sqrt{2\pi}\sigma_1} \times (1-p_{c_2}) \times \lambda_f e^{-\lambda_f} \times \cdots \times p_{c_N} \frac{e^{-(s_N-h_N)^2/2\sigma_N^2}}{\sqrt{2\pi}\sigma_N}.$$

In the most general case, we proceed as follows. Let

$N \equiv$ Number of cuts in the hypothesis $\mathcal{H}$.

$h_i \equiv$ The ith cut location on $\mathcal{H}$.

$M_j \equiv$ Number of cuts in the data D_j.

$K_j \equiv$ Number of possible alignments of the data/evidence D_j against the hypothesis $\mathcal{H}$ (or its reversal, the flipped alignment $\mathcal{H}^R$).

$s_{ijk} \equiv$ The cut location in D_j matching the cut h_i in $\mathcal{H}$, given the alignment A_{jk}. In case such a match occurs, this event is denoted by an indicator variable m_{ijk} taking the value 1.

$m_{ijk} \equiv$ An indicator variable, taking the value 1 iff the cut s_{ijk} in D_j matches a cut h_i in the hypothesis $\mathcal{H}$, given the alignment A_{jk}. It takes the value 0, otherwise.

$F_{jk} \equiv$ Number of false (non-matching) cuts in the data D_j for alignment A_{jk}, that do not match any cut in the hypothesis $\mathcal{H}$. Thus

$$F_{jk} = M_j - \sum_{i=1}^{N} m_{ijk}$$

Number of missing cuts is thus

$$\sum_{i=1}^{N} (1 - m_{ijk}) = N - \sum_{i=1}^{N} m_{ijk}.$$

By an abuse of notation, we may omit the indices j and k, if from the context it can be uniquely determined which data D_j and alignment A_{jk} are being referred to.

Note that a fixed alignment A_{jk} can be uniquely described by marking the cuts on D_j by the labels T (for true cut) and F (for false cut) and by further augmenting each true cut by the identity of the cut h_i of the hypothesis $\mathcal{H}$. From this information, m_{ijk}, s_{ijk}, F_{jk}, etc. can all be uniquely determined. Let the cuts of D_j be $(s_1, s_2, \ldots, s_{M_j})$. Let the event E_i denote the situation that there is a cut in the infinitesimal interval $(s_i - \Delta x/2, s_i + \Delta x/2)$.

Thus we have:

$$\begin{aligned}
&\mathbb{P}\mathrm{r}[D_j^{(k)}|\mathcal{H},\mathrm{good}]\Delta x_1\cdots\Delta x_{M_j}\\
&\quad= \mathbb{P}\mathrm{r}[D_j^{(k)}|\mathcal{H},\mathrm{good}](\Delta x)^{M_j}\\
&\quad= prob[E_1,\ldots,E_{M_j},A_{jk}|\mathcal{H},\mathrm{good}]\\
&\quad= prob[E_1,\ldots,E_{M_j},A_{jk}|m_{ijk},M_j,\mathcal{H},\mathrm{good}]\times prob[m_{ijk},M_j|\mathcal{H},\mathrm{good}]\\
&\quad= prob[E_1,A_{jk}|m_{ijk},M_j,\mathcal{H},\mathrm{good}]\times prob[E_2,A_{jk}|E_1,m_{ijk},M_j,\mathcal{H},\mathrm{good}]\\
&\qquad\quad\times\cdots\times prob[E_\alpha,A_{jk}|E_1,\ldots,E_{\alpha-1},m_{ijk},M_j,\mathcal{H},\mathrm{good}]\times\cdots\\
&\qquad\quad\times prob[E_{M_j},A_{jk}|E_1,\ldots,E_{M_j-1},m_{ijk},M_j,\mathcal{H},\mathrm{good}]\\
&\qquad\qquad\times prob[m_{ijk},M_j|\mathcal{H},\mathrm{good}]
\end{aligned}$$

Note the following:

1.

$$\begin{aligned}
prob[m_{ijk},M_j|\mathcal{H},\mathrm{good}] &= \left[\prod_{i=1}^{N}(p_{c_i}m_{ijk}+(1-p_{c_i})(1-m_{ijk}))\right]\times e^{-\lambda_f}{\lambda_f}^{F_{jk}}/F_{jk}!\\
&= \left[\prod_{i=1}^{N}p_{c_i}^{m_{ijk}}\ (1-p_{c_i})^{(1-m_{ijk})}\right]\times e^{-\lambda_f}{\lambda_f}^{F_{jk}}/F_{jk}!
\end{aligned}$$

2. For the event E_α there are two possible situations to be considered:

   (a) s_α is a false cut and the number of false cuts among s_1, …, $s_{\alpha-1}$ is β.

   $$prob[E_\alpha,A_{jk}|E_1,\ldots,E_{\alpha-1},m_{ijk},M_j,\mathcal{H},\mathrm{good}]=(F_{jk}-\beta)\Delta x.$$

   (b) $s_\alpha=s_{ijk}$ is a true cut and h_i is the cut in $\mathcal{H}$ associated with it.

   $$prob[E_\alpha,A_{jk}|E_1,\ldots,E_{\alpha-1},m_{ijk},M_j,\mathcal{H},\mathrm{good}]=\frac{e^{-(s_{ijk}-h_i)^2/2\sigma_i^2}}{\sqrt{2\pi}\sigma_i}\Delta x.$$

   Thus,

   $$\begin{aligned}
   &prob[E_1,\ldots,E_{M_j},A_{jk}|m_{ijk},M_j,\mathcal{H},\mathrm{good}]\\
   &\quad= \prod_{i=1}^{N}\left(\frac{e^{-(s_{ijk}-h_i)^2/2\sigma_i^2}}{\sqrt{2\pi}\sigma_i}\Delta x\right)^{m_{ijk}}\times F_{jk}!(\Delta x)^{F_{jk}}\\
   &\quad= F_{jk}!\prod_{i=1}^{N}\left(\frac{e^{-(s_{ijk}-h_i)^2/2\sigma_i^2}}{\sqrt{2\pi}\sigma_i}\right)^{m_{ijk}}(\Delta x)^{M_j}
   \end{aligned}$$

Putting it all together:

$$\begin{aligned}&\mathbb{P}\mathrm{r}[D_j^{(k)}|\mathcal{H},\mathrm{good}]\\ &= \left[\prod_{i=1}^{N}\left(p_{c_i}\frac{e^{-(s_{ijk}-h_i)^2/2\sigma_i^2}}{\sqrt{2\pi}\sigma_i}\right)^{m_{ijk}}(1-p_{c_i})^{(1-m_{ijk})}\right]\times e^{-\lambda_f}{\lambda_f}^{F_{jk}}. \qquad (1)\end{aligned}$$

By an identical argument we see that the only alignments relevant for the bad molecules correspond to the situation when all cuts in D_j are labeled false, and for each of two such alignments,

$$\mathbb{P}\mathrm{r}[D_j^{(k)}|\mathcal{H},\mathrm{bad}] = e^{-\lambda_n}{\lambda_n}^{M_j}.$$

The log-likelihood can then be computed as follows:

$$\mathcal{L} \equiv \sum_j \log \mathbb{P}\mathrm{r}[D_j|\mathcal{H}].$$

Thus,

$$\begin{aligned}\mathcal{L} &= \sum_j \log\left[p_b e^{-\lambda_n}{\lambda_n}^{M_j} + \frac{(1-p_b)}{2}\sum_k \mathbb{P}\mathrm{r}_{jk}\right]\\ &= \sum_j \log[p_b e_j + (1-p_b)d_j]\\ &\qquad \text{where, by definition, } e_j \equiv e^{-\lambda_n}{\lambda_n}^{M_j},\\ &\qquad \text{and } d_j \equiv (\sum_k \mathbb{P}\mathrm{r}_{jk})/2.\end{aligned}$$

4.2 Prior Distribution in the Hypotheses Space

In the model, we shall use an extremely simple distribution on the prior $\mathbb{P}\mathrm{r}[\mathcal{H}]$ that only depends on the number of restriction sites, N and not any other parameters. Implicitly, we assume that all hypotheses with same number of cuts are equi-probable, independent of the cut location.

Given a k-cutter enzyme (e.g., normally six-cutters like *EcoR* I in our case), the probability that it cuts at any specific site in a sufficiently long clone is given by

$$p_e = \left(\frac{1}{4}\right)^k.$$

Thus if the clone is of length G bps and if we denote by $\lambda_e = Gp_e$ (the expected number of restriction sites in the clone), then the probability that the clone has exactly N restriction cuts is given by

$$prob[\#\text{ restriction sites } = N|\text{ enzyme, } e \text{ and clone of length } G] \approx e^{-\lambda_e}\frac{\lambda_e^N}{N!}.$$

The preceding computation is based on the assumption that all four bases $\in \{A, T, C, G\}$ occur in the clone with equal probability $\frac{1}{4}$. However, as it is known,[BSW84], human genome is CG-poor (i.e., $\mathbb{P}\mathrm{r}[C] + \mathbb{P}\mathrm{r}[G] = 0.32 < \mathbb{P}\mathrm{r}[A] + \mathbb{P}\mathrm{r}[T] = 0.68$), a more realistic model can use a better estimation for p_e:

$$p_e = (0.16)^{\#CG}(0.34)^{\#AT},$$

where $\#CG$ denotes the number of C or G in the restriction sequence for the enzyme and similarly, $\#AT$ denotes the number of A or T in the restriction sequence.

4.3 Local Search Algorithm

In order to find the most plausible restriction map, we shall optimize the cost function derived earlier, with respect to the following parameters:

$$\begin{aligned}
\text{Cut Sites} &= h_1, h_2, \ldots, h_N, \\
\text{Cut Rates} &= p_{c_1}, p_{c_2}, \ldots, p_{c_N}, \\
\text{Std. Dev. of Cut Sites} &= \sigma_1, \sigma_2, \ldots, \sigma_N, \\
\text{Auxiliary Parameters} &= p_b, \lambda_f \text{ and } \lambda_n.
\end{aligned}$$

Let us denote any of these parameters by θ. Thus, we need to solve the equation

$$\frac{\partial \mathcal{L}}{\partial \theta} = 0,$$

to find an extremal point of $\mathcal{L}$ with respect to the parameter θ.

4.3.1 Case 1: $\theta \to p_b$

Taking the first partial derivative, we get

$$\frac{\partial \mathcal{L}}{\partial p_b} = \sum_j \frac{(e_j - d_j)}{p_b e_j + (1 - p_b) d_j}. \tag{2}$$

Taking the second partial derivative, we get

$$\frac{\partial^2 \mathcal{L}}{\partial {p_b}^2} = -\sum_j \frac{(e_j - d_j)^2}{[p_b e_j + (1 - p_b) d_j]^2}. \tag{3}$$

Accordingly, $\mathcal{L}$ can now be easily optimized iteratively to estimate the best value of p_b, by means of the following applications of the Newton's equation:

$$p_b := p_b - \frac{\partial \mathcal{L}/\partial p_b}{\partial^2 \mathcal{L}/\partial {p_b}^2}.$$

4.3.2 Case 2: $\theta \to \lambda_n$

This parameter is simply estimated to be the average number of cuts. Note that,

$$\frac{\partial \mathcal{L}}{\partial \lambda_n} = \sum_j \frac{p_b e_j (M_j/\lambda_n - 1)}{p_b e_j + (1-p_b) d_j}$$

should be zero at the local maxima. Thus a good approximation is obtained by taking

$$\sum_j \left(\frac{M_j}{\lambda_n} - 1\right) \approx 0,$$

leading to the update rule

$$\lambda_n := \frac{\sum_j M_j}{\sum_j 1} = \frac{\sum_j M_j}{\text{Total number of molecules}}.$$

Thus λ_n is simply the average number of cuts per molecule.

4.3.3 Case 3: $\theta \to h_i$, p_{c_i}, σ_i $(i = 1, \ldots, N)$, or λ_f

Unlike in the previous two cases, these parameters are in the innermost section of our probability density expression and computing any of these gradients will turn out to be computationally comparable to evaluating the entire probability density.

In this case,

$$\frac{\partial \mathcal{L}}{\partial \theta} = \sum_j \frac{1}{\mathbb{P}\mathrm{r}_j} \left(\frac{1-p_b}{2} \sum_k \mathbb{P}\mathrm{r}_{jk} \chi_{jk}(\theta)\right), \quad \text{where } \mathbb{P}\mathrm{r}_j \equiv \mathbb{P}\mathrm{r}[D_j|\mathcal{H}] \quad \text{and}$$

where

$$\begin{aligned} \chi_{jk}(\theta) \equiv & \left[\frac{F_{jk}}{\lambda_f}\frac{\partial \lambda_f}{\partial \theta} - \frac{\partial \lambda_f}{\partial \theta}\right] \\ & + \sum_{i=1}^{N} \left[\frac{m_{ijk}}{p_{c_i}}\frac{\partial p_{c_i}}{\partial \theta} - \frac{1-m_{ijk}}{1-p_{c_i}}\frac{\partial p_{c_i}}{\partial \theta}\right] \\ & + \sum_{i=1}^{N} m_{ijk} \left[\frac{\partial}{\partial \theta}\left(\frac{-(s_{ijk}-h_i)^2}{2\sigma_i^2}\right) - \frac{1}{\sigma_i}\frac{\partial \sigma_i}{\partial \theta}\right]. \end{aligned}$$

For convenience, now define

$$\begin{aligned} \pi_{jk} \quad & \equiv \quad \left(\frac{1-p_b}{2}\right) \frac{\mathbb{P}\mathrm{r}_{jk}}{\mathbb{P}\mathrm{r}_j} \\ & \equiv \quad \text{Relative probability density of the alignment } A_{jk} \text{ for data item } D_j. \end{aligned}$$

Thus, our earlier formula for $\frac{\partial \mathcal{L}}{\partial \theta}$ now simplifies to

$$\frac{\partial \mathcal{L}}{\partial \theta} = \sum_j \sum_k \pi_{jk}\, \chi_{jk}(\theta).$$

Before, examining the updating formula for each parameter optimization, we shall introduce the following notations for future use. The quantities defined below can be efficiently accumulated for a fixed value of the set of parameters.

$$
\begin{array}{lcllcl}
\Psi_{0i} & \equiv & \sum_j \sum_k \pi_{jk} m_{ijk} & \equiv & \text{Expected number of cuts matching } h_i \\
\Psi_{1i} & \equiv & \sum_j \sum_k \pi_{jk} m_{ijk} s_{ijk} & \equiv & \text{Sum of cut locations matching } h_i \\
\Psi_{2i} & \equiv & \sum_j \sum_k \pi_{jk} m_{ijk} s_{ijk}^2 & \equiv & \text{Sum of square of cut locations matching } h_i \\
\mu_g & \equiv & \sum_j \sum_k \pi_{jk} & \equiv & \text{Expected number of ``good'' molecules} \\
\gamma_g & \equiv & \sum_j \sum_k \pi_{jk} M_j & \equiv & \text{Expected number of cuts in ``good'' molecules}
\end{array}
$$

We note here that Ψ's can all be computed efficiently using a simple updating rule that modifies the values with one data item D_j (molecule) at a time. This rule can then be implemented using a Dynamic Programming recurrence equation (described later).

Case 3A: $\theta \to h_i$ Note that,

$$
\begin{aligned}
& \theta \equiv h_i \\
\Rightarrow \quad & \chi_{jk}(h_i) = m_{ijk}(s_{ijk} - h_i)/\sigma_i^2 \\
\Rightarrow \quad & \frac{\partial \mathcal{L}}{\partial h_i} = \sum_j \sum_k \pi_{jk} m_{ijk}(s_{ijk} - h_i)/\sigma_i^2.
\end{aligned}
$$

Thus,

$$\frac{\partial \mathcal{L}}{\partial h_i} = \frac{1}{\sigma_i^2}\left(\Psi_{1i} - h_i \Psi_{0i}\right).$$

Although, Ψ's depend on the location h_i, they vary rather slowly as a function of h_i. Hence a feasible update rule for h_i is

$$h_i := \frac{\Psi_{1i}}{\Psi_{0i}}. \tag{4}$$

Thus the updated value of h_i is simply the "average expected value" of all the s_{ijk}'s that match the current value of h_i.

Case 3B: $\theta \to p_{c_i}$ Note that,

$$
\begin{aligned}
& \theta \equiv p_{c_i} \\
\Rightarrow \quad & \chi_{jk}(p_{c_i}) = \frac{m_{ijk}}{p_{c_i}} - \frac{1 - m_{ijk}}{1 - p_{c_i}} \\
\Rightarrow \quad & \frac{\partial \mathcal{L}}{\partial p_{c_i}} = \sum_j \sum_k \pi_{jk} \frac{m_{ijk}}{p_{c_i}} - \frac{1 - m_{ijk}}{1 - p_{c_i}}.
\end{aligned}
$$

Thus,

$$\frac{\partial \mathcal{L}}{\partial p_{c_i}} = \frac{\Psi_{0i}}{p_{c_i}} - \frac{\mu_g - \Psi_{0i}}{1 - p_{c_i}}.$$

Again, arguing as before, we have the following feasible update rule for p_{c_i}

$$p_{c_i} := \frac{\Psi_{0i}}{\mu_g}. \tag{5}$$

Thus p_{c_i} is just the fraction of the good molecules that have a matching cut at the current value of h_i.

Case 3C: $\theta \to \sigma_i$ Note that,

$$\begin{aligned}
& \theta \equiv \sigma_i \\
\Rightarrow \quad & \chi_{jk}(\sigma_i) = m_{ijk}\left(\frac{(s_{ijk}-h_i)^2}{\sigma_i^3} - \frac{1}{\sigma_i}\right) \\
\Rightarrow \quad & \frac{\partial \mathcal{L}}{\partial \sigma_i} = \sum_j \sum_k \pi_{jk} m_{ijk}\left(\frac{(s_{ijk}-h_i)^2}{\sigma_i^3} - \frac{1}{\sigma_i}\right).
\end{aligned}$$

Thus,

$$\frac{\partial \mathcal{L}}{\partial \sigma_i} = \frac{1}{\sigma_i^3}\left(\Psi_{2i} - 2h_i\Psi_{1i} + h_i^2\Psi_{0i} - \sigma_i^2\Psi_{0i}\right).$$

Thus, we have the following feasible update rule for σ_i^2

$$\sigma_i^2 := \frac{(\Psi_{2i} - 2h_i\Psi_{1i} + h_i^2\Psi_{0i})}{\Psi_{0i}}.$$

Using the estimate for h_i (equation (4)), we have

$$\sigma_i^2 := \frac{\Psi_{2i}}{\Psi_{0i}} - \left(\frac{\Psi_{1i}}{\Psi_{0i}}\right)^2. \tag{6}$$

This is simply the variance of all the s_{ijk}'s that match the current value of h_i.

Case 3D: $\theta \to \lambda_f$ Note that,

$$\begin{aligned}
& \theta \equiv \lambda_f \\
\Rightarrow \quad & \chi_{jk}(\lambda_f) = \frac{F_{jk}}{\lambda_f} - 1 = \frac{M_j - \sum_i m_{ijk}}{\lambda_f} - 1 \\
\Rightarrow \quad & \frac{\partial \mathcal{L}}{\partial \lambda_f} = \sum_j \sum_k \pi_{jk}\left(\frac{M_j - \sum_i m_{ijk}}{\lambda_f} - 1\right) \\
& \qquad = \frac{\gamma_g - \sum_i \Psi_{0i}}{\lambda_f} - \mu_g.
\end{aligned}$$

Thus, we have the following feasible update rule for λ_f

$$\lambda_f := \frac{\gamma_g}{\mu_g} - \sum_i \frac{\Psi_{0i}}{\mu_g}. \tag{7}$$

This is simply the average number of unmatched cuts per "good" molecule. (Note that the molecules are already normalized to unit length.)

Case 3E: $\theta \to p_c = p_{c_1} = \cdots = p_{c_N}$ **(Constrained)** Note that,

$$\frac{\partial \mathcal{L}}{\partial p_c} = \sum_j \sum_k \sum_{i=1}^{N} \pi_{jk} \left(\frac{m_{ijk}}{p_c} - \frac{1 - m_{ijk}}{1 - p_c} \right)$$

Thus the update equation for this case is:

$$p_c := \frac{\sum_i \Psi_{0i}/N}{\mu_g}. \tag{8}$$

Case 3F: $\theta \to \sigma = \sigma_1 = \cdots = \sigma_N$ **(Constrained)** Note that,

$$\frac{\partial \mathcal{L}}{\partial \sigma} = \sum_j \sum_k \sum_{i=1}^{N} \pi_{jk} m_{ijk} \left(\frac{(s_{ijk} - h_i)^2}{\sigma^3} - \frac{1}{\sigma} \right).$$

Thus the update equation for this case is:

$$\sigma^2 := \frac{\sum_i (\Psi_{2i} - \Psi_{1i}^2/\Psi_{0i})}{\sum_i \Psi_{0i}}. \tag{9}$$

4.4 Update Algorithm: Dynamic Programming

In each update step, we need to compute the new values of the parameters based on the old values of the parameters, which affect the "moment functions:" Ψ_{0i}, Ψ_{1i}, Ψ_{2i}, μ_g and γ_g. For the ease of expressing the computation, we shall use additional auxiliary expressions as follows:

$$\left.\begin{array}{lcl} \mathtt{P}_j & \equiv & \sum_k \left(\frac{\mathbb{P}\mathrm{r}_{jk}}{e^{-\lambda_f}} \right) \\ \mathtt{W}_{ij} & \equiv & \sum_k \left(\frac{\mathbb{P}\mathrm{r}_{jk} m_{ijk}}{e^{-\lambda_f}} \right) \\ \mathtt{SUM}_{ij} & \equiv & \sum_k \left(\frac{\mathbb{P}\mathrm{r}_{jk} m_{ijk} s_{ijk}}{e^{-\lambda_f}} \right) \\ \mathtt{SQ}_{ij} & \equiv & \sum_k \left(\frac{\mathbb{P}\mathrm{r}_{jk} m_{ijk} s_{ijk}^2}{e^{-\lambda_f}} \right) \end{array}\right\} \tag{10}$$

One motivation for this formulation is to avoid having to compute $e^{-\lambda_f}$ repeatedly, since this is a relatively expensive computation. Note that, the original moment function can now be computed as follows:

$$\left.\begin{array}{lcl} \mathbb{P}\mathrm{r}_j & = & \left(\frac{1-p_b}{2}\right) e^{-\lambda_f} \times \mathtt{P}_j + p_b e_j \\ \Psi_{0i} & = & \left(\frac{1-p_b}{2}\right) e^{-\lambda_f} \sum_j \frac{\mathtt{W}_{ij}}{\mathbb{P}\mathrm{r}_j} \\ \Psi_{1i} & = & \left(\frac{1-p_b}{2}\right) e^{-\lambda_f} \sum_j \frac{\mathtt{SUM}_{ij}}{\mathbb{P}\mathrm{r}_j} \\ \Psi_{2i} & = & \left(\frac{1-p_b}{2}\right) e^{-\lambda_f} \sum_j \frac{\mathtt{SQ}_{ij}}{\mathbb{P}\mathrm{r}_j} \\ \mu_g & = & \left(\frac{1-p_b}{2}\right) e^{-\lambda_f} \sum_j \frac{\mathtt{P}_j}{\mathbb{P}\mathrm{r}_j} \\ \gamma_g & = & \left(\frac{1-p_b}{2}\right) e^{-\lambda_f} \sum_j \frac{M_j \mathtt{P}_j}{\mathbb{P}\mathrm{r}_j} \end{array}\right\} \tag{11}$$

Finally,

$$\mathbb{P}\mathrm{r}[\mathcal{D}|\mathcal{H}] = \prod_j \mathbb{P}\mathrm{r}_j.$$

The definitions for $\mathtt{P}_j$, $\mathtt{W}_{ij}$, $\mathtt{SUM}_{ij}$ and $\mathtt{SQ}_{ij}$ involve *all alignments* between each data element D_j and the hypothesis $\mathcal{H}$. This number is easily seen to be exponential in the number of cuts N in the hypothesis $\mathcal{H}$, *even if one excludes such physically impossible alignments as the ones involving cross-overs* (i.e., alignments in which the order of cuts in $\mathcal{H}$ and D_j are different). First, consider $\mathtt{P}_j$:

$$\begin{aligned}
\mathtt{P}_j &\equiv \sum_k \left(\frac{\mathbb{P}\mathrm{r}_{jk}}{e^{-\lambda_f}}\right) \\
&= \sum_k \left\{ \prod_{i=1}^{N} \left(p_{c_i} \frac{e^{-(h_i - s_{ijk})^2/2\sigma_i^2}}{\sqrt{2\pi}\sigma_i} \right)^{m_{ijk}} \times \prod_{i=1}^{n} (1-p_{c_i})^{1-m_{ijk}} \times {\lambda_f}^{F_j k} \right\}
\end{aligned}$$

Next we shall describe recurrence equations for computing the values for all alignments efficiently. The set of alignments computed are for the cuts $\{1, \ldots, M_j\}$ of D_j mapped against the hypothesized cuts $\{1, \ldots, N\}$. We define the recurrence equations in terms of

$$\mathtt{P}_{q,r} \equiv \mathtt{P}_j(s_q, \ldots, s_{M_j}; h_r, \ldots, h_N),$$

which is the probability density of all alignments for the simpler problem in which cuts s_1, ..., s_{q-1} are missing in the data D_j and the cuts h_1, ..., h_{r-1} are missing in the hypothesis $\mathcal{H}$.

Then, clearly

$$\begin{aligned}
\mathtt{P}_j &\equiv \mathtt{P}_{1,1} \\
\mathtt{P}_{q,r} &\equiv \lambda_f \mathtt{P}_{q+1,r} + \sum_{t=r}^{N} \mathtt{P}_{q+1,t+1} \left\{ \prod_{i=r}^{t-1} (1-p_{c_i}) \right\} p_{c_t} \frac{e^{-(h_t - s_q)^2/2\sigma_t^2}}{\sqrt{2\pi}\sigma_t}, \qquad (12) \\
&\text{where } 1 \le q \le M_j \text{ and } 1 \le r \le N+1.
\end{aligned}$$

This follows from a nested enumeration of all possible alignments. The recurrence terminates in $\mathtt{P}_{M_j+1,r}$, which represents P_j if all cuts in D_j were missing and cuts h_1, ..., h_{r-1} in $\mathcal{H}$ were missing:

$$\mathtt{P}_{M_j,r} = \prod_{i=r}^{N} (1-p_{c_i}). \qquad (13)$$

Thus the total number of terms $\mathtt{P}_{q,r}$ to be computed is bounded from above by $(M_j+1)(N+1)$ where M_j is the number of cuts in data molecule D_j and N is the number cuts in $\mathcal{H}$. Each term can be computed in descending order of q and r using equations (12) and (13). The time complexity associated with the computation of $\mathtt{P}_{q,r}$ is $O(N-r)$ in terms of the arithmetic operations.

Note also that the equation (12) can be written in the following alternative form:

$$\mathtt{P}_j \equiv \mathtt{P}_{1,1}$$

$$\mathtt{P}_{q,r} \equiv \lambda_f \mathtt{P}_{q+1,r} + \mathtt{P}_{q+1,r+1} p_{c_t} \frac{e^{-(h_t - s_q)^2/2\sigma_t^2}}{\sqrt{2\pi}\sigma_t} + (1-p_{c_r})\Big\{\mathtt{P}_{q,r+1} - \lambda_f \mathtt{P}_{q+1,r+1}\Big\}, \quad (14)$$
$$\text{where } 1 \le q \le M_j \text{ and } 1 \le r \le N+1.$$

Thus, by computing $\mathtt{P}_{q,r}$ in descending order of r, *only two new terms* [and *one new product* $(1-p_{c_r})$ in equation (14)] needs be to be computed for each $\mathtt{P}_{q,r}$. With this modification, the overall time complexity reduces to $O(M_j N)$.

The complexity can be further improved by taking advantage of the fact that the exponential term is negligibly small unless h_t and s_q are sufficiently close (e.g., $|h_t - s_q| \le 3\sigma_t$). For any given value of q, only a small number of h_t will be close to s_q. For a desired finite precision *only a small constant fraction* of h_t's will be sufficiently close to s_q to require that the term with the exponent be included in the summation[3].

Note however that even with this optimization in the computation for equation (12), the computation of $\mathtt{P}_{q,r}$ achieves no asymptotic improvement in the time complexity, since $\mathtt{P}_{q,r}$ with consecutive r can be computed with *only two new terms*, as noted earlier. However, for any given q, only for a few r values are both of these additional terms non-negligible. The range of r values (say, between $r_{\min}$ and $r_{\max}$) for which the new terms with $e^{-(h_r-s_q)^2/2\sigma_t^2}$ is significant can be precomputed in a table indexed by $q = 1, \ldots, M_j$. For $r > r_{\max}$ all terms in the summation are negligible. For $r < r_{\min}$ the new exponential term referred to previously is negligible. In both cases, the expression for $\mathtt{P}_{q,r}$ can be simplified:

$$\mathtt{P}_{q,r} = \begin{cases} \lambda_f \mathtt{P}_{q+1,r}, & \text{if } r > r_{\max}[q]; \\ \lambda_f \mathtt{P}_{q+1,r} + (1-p_{c_r})(\mathtt{P}_{q,r+1} - \lambda_f \mathtt{P}_{q+1,r+1}), & \text{if } r < r_{\min}[q]. \end{cases} \quad (15)$$

Since both $r_{\min}[q]$ and $r_{\max}[q]$ are monotonically nondecreasing functions of q, the (q, r) space divides as shown in figure 3. Of course, the block diagonal pattern need not be as regular as shown and will differ for each data molecule D_j.

Note again that our ultimate object is to compute $\mathtt{P}_{1,1}$. Terms $\mathtt{P}_{q,r+1}$ with $r > r_{\max}[q]$, cannot influence any term $\mathtt{P}_{q',r'}$ with $r' \le r$ (see equation (12)). Therefore, any term $\mathtt{P}_{q,r+1}$ with $r > r_{\max}[q]$ cannot influence $\mathtt{P}_{1,1}$ as is readily seen by a straightforward inductive argument. Therefore, all such terms need not be computed at all.

For $r < r_{\min}[q]$, these terms are required but need not be computed since they always satisfy the following identity:

$$\mathtt{P}_{q,r} = (1-p_{c_r})\mathtt{P}_{q,r+1}, \quad r < r_{\min}[q].$$

This follows from equation (13) and (15) by induction on q. These terms can then be generated on demand when the normal recurrence (equation (12)) is computed and whenever a term $\mathtt{P}_{q+1,r}$ is required for which $r < r_{\min}[q+1]$, provided terms are processed in descending order of r.

Thus, the effective complexity of the algorithm is $O(M_j(r_{\max} - r_{\min} + 2))$. Since $r_{\max} - r_{\min} + 2$ is proportional for a given precision to $\lceil(\sigma N + 1)\rceil$, (where σ is an upper bound on all the σ_t values) we see that the time complexity for a single molecule D_j is $O(\sigma M_j N)$.

[3]In practice, even a precision of 10^{-10} will only requires 3–5 terms to be included with σ around 1%.

Summing over all molecules D_j the total time complexity is $O(\sigma MN)$, where $M = \sum_j M_j$. The space complexity is trivially bounded by $O(M_{\max} N)$ where $M_{\max} = \max_j M_j$.

Essentially the same recurrence equations can be used to compute $\mathtt{W}_{ij}$, $\mathtt{SUM}_{ij}$ and $\mathtt{SQ}_{ij}$, since these $3N$ quantities sum up the same probability densities $\mathbb{P}\mathrm{r}_{jk}$ weighted by m_{ijk}, $m_{ijk}s_{ijk}$ or $m_{ijk}s_{ijk}^2$ respectively. The difference is that the termination of the recurrence (cf equation (13)) is simply $P_{M_j+1,r} = 0$, whereas the basic recurrence equation (cf equation (12)) contains an additional term corresponding to the m_{ijk} times the corresponding term in the recurrence equation. For example:

$$\begin{aligned}
\mathtt{SUM}_{ij} &\equiv \mathtt{SUM}_{i,1,1} \\
\mathtt{SUM}_{i,q,r} &\equiv \lambda_f \mathtt{SUM}_{i,q+1,r} + \sum_{t=r}^{N} \mathtt{SUM}_{i,q+1,t+1} \left\{ \prod_{j=r}^{t-1} (1-p_{c_j}) \right\} p_{c_t} \frac{e^{-(h_t - s_q)^2/2\sigma_t^2}}{\sqrt{2\pi}\sigma_t} \\
&\quad + \mathbb{I}_{i \geq r}\ s_q \mathtt{P}_{q+1,i+1} \left\{ \prod_{j=r}^{i-1} (1-p_{c_j}) \right\} p_{c_i} \frac{e^{-(h_i - s_q)^2/2\sigma_i^2}}{\sqrt{2\pi}\sigma_i}, \qquad (16) \\
&\quad \text{where } 1 \leq q \leq M_j \text{ and } 1 \leq r \leq N+1.
\end{aligned}$$

Note that the new term is only present[4] if $i \geq r$, and as before need only be computed if the corresponding exponent is significant, i.e., i lies between $r_{\min}[q]$ and $r_{\max}[q]$. This term is the only nonzero input term in the recurrence since the terminal terms are zero. This recurrence is most easily derived by noting (from equations (1) and (10)) that the sum of products form of $\mathtt{SUM}_{ij}$ can be derived from that of $\mathtt{P}_j$ by multiplying each product term with $h_i - s_q$ in any exponent by s_q, and deleting any term without h_i in the exponent. Since each product term contains at most one exponent with h_i, this transformation can also be applied to the recurrence form for $\mathtt{P}_j$ (equation (12)), which is just a different factorization of the original sum of products form. The result is equation (16). The corresponding derivation for $\mathtt{W}_{ij}$ and $\mathtt{SQ}_{ij}$ is the same except that the s_q is replaced by 1 or s_q^2 respectively. If the recurrences for these $3N$ quantities are computed in parallel with the probability density $\mathtt{P}_j$, the cost of the extra term is negligible, so the overall cost of computing both the probability density $\mathtt{P}_j$ and its gradients is $O(\sigma MN^2)$. The cost of conversion equations (11) is also negligible in comparison. Moreover this can be implemented as a vectorized version of the basic recurrence with vector size $3N+1$ to take advantage of either vector processors or superscalar pipelined processors. We note in passing that if $3N$ is significantly greater than the average width σM of the dynamic programming block diagonal matrix shown in figure 3, then a standard strength reduction can be applied to the vectorized recurrence equations trading the $3N$ vector size for a $\sigma N + 1$ vector size and resulting in an alternate complexity of $O(\sigma^2 MN^2)$. We have not tried implementing this version since it is much harder to code and the gain is significant only when $\sigma << 1$. Note that the gradient must

[4]The indicator function $\mathbb{I}_{i\geq r} \equiv (i \geq r ? 1 : 0)$ is a shorthand for

$$\begin{cases} 1, & \text{if } i \geq r; \\ 0, & \text{otherwise.} \end{cases}$$

be computed a number of times (typically 10-20 times) for the parameters to converge to a local maxima.

We note that similar ideas have been explored in the work of G. Churchill [Chu97] in the context of sequences assumed to be related by descent from a common ancestor. However, it is not clear whether the general framework proposed by Churchill applies to our formulation.

4.5 Global Search Algorithm

Recall that our prior distribution $\mathbb{P}\mathrm{r}[\mathcal{D}|\mathcal{H}]$ is multimodal and the local search based on the gradients by itself cannot evaluate the best value of the parameters. Instead, we must rely on some sampling method to find points in the parameter space that are likely to be near the global maxima. Furthermore, examining the parameter space, we notice that the parameters corresponding to the number and locations of restriction sites present the largest amount of multimodal variability and hence the sampling may be restricted to just $\bar{h} = (N; h_1, h_2, \ldots, h_N)$. The conditional observation probability density $\mathbb{P}\mathrm{r}[\mathcal{D}|\mathcal{H}]$ can be evaluated pointwise in time $O(\sigma M N)$ and the nearest local maxima located in time $O(\sigma M N^2)$, though there is no efficient way to sample all local maxima exhaustively.

Thus, our global search algorithm will proceed as follows: we shall first generate a set of samples $(\bar{h}_1, \bar{h}_2, \bar{h}_3, \ldots)$; these points are then used to begin a gradient search for the nearest maxima and provide hypotheses $(\mathcal{H}_1, \mathcal{H}_2, \mathcal{H}_3, \ldots)$; the hypotheses are then ranked in terms of their posterior probability density $\mathbb{P}\mathrm{r}[\mathcal{H}|\mathcal{D}]$ (whose relative values also lead to the quality measure for each hypothesis) and the one (or more) leading to maximal posterior probability density is presented as the final answer.

However, even after restricting the sampling space, the high dimension of the space makes the sampling task daunting. Even if the space is discretized (for instance, each $h_i \in \{0, 1/200, \ldots, j/200, \ldots, 1\}$), there are still far too many sample points (200^N) even for a small number of cuts (say, $N = 8$). However, the efficiency can be improved if we accept an approximate solution. We shall rely on following two approaches (and their combination):

1. We may use approximate Bayesian probability densities in conjunction with a branch and bound algorithm to reject a large fraction of the samples without further local analysis;

2. We may use an approximate posterior distribution for the location of the cut sites in conjunction with a Monte Carlo approach to generate samples that are more likely to succeed in the local analysis.

One can also combine the two methods: for instance, we can use the first approach to generate the best hypotheses with small (say, 5) number of cuts and use it to improve the approximate posterior to be used in the second approach. Note also that, if the data quality is "good," rather simple versions of the heuristics (for global search) lead to greedy algorithms that yield good results quite fast. However, we shall only describe the first approach here and postpone the discussion of the second heuristics to a sequel.

For the present the parameter N is searched in strictly ascending order. This means one first evaluates the (single) map with no cuts, then applies global and gradient search to locate the best map with 1 cut, then the best map with 2 cuts etc. One continues until the score of the best map of N cuts is significantly worse than the best map of 0 ... $N-1$ cuts.

4.5.1 Approximating Bayesian Probability Densities

The global search for a particular N uses an approximate Bayesian probability density with a scoring function that is amenable to efficient branch-and-bound search. Observe that good scores for some molecule D_j, basically requires that as many cut locations $s_{1j}, \ldots, s_{M_j,j}$ as possible must line up close to $h_1, h_2, \ldots, h_N$ in one of the two orientations. This means that any subset of the true map $h_1, h_2, \ldots, h_m$ $(m < N)$ will score better than most other maps of size m, assuming that the digest rate is equal $(p_c = p_{c_1} = \cdots = p_{c_N})$. Note that for physical reasons, the variation among the digest rates is quite small; thus, our assumption is valid and permits us to explicitly constrain these parameters to be the same. For example, if $(h_1, h_2, \ldots, h_N)$ is the correct map, one expects maps with single cuts located at $[h_i]$ $(1 \leq i \leq N)$ to score about equally well in terms of the Bayesian probability density. Similarly, maps with two cuts located at pairs of $[h_i, h_j]$ $(1 \leq i < j \leq N)$ score about equally well and better than arbitrarily chosen two cut maps. Furthermore, the pair-cut probability densities are more robust than the single cut probability densities with respect to the presence of false cuts, hence, less likely to score maps with cuts in other than the correct locations. Hence an approximate score function used for a map $(h_1, h_2, \ldots, h_N)$ is the smallest probability density for any pair map $[h_i, h_j]$ which is a subset of $(h_1, h_2, \ldots, h_N)$. These pair map probability densities can be precomputed for every possible pair $([h_i, h_j])$ if h_i, h_j are forced to have only K values along some finite sized grid, for example at exact multiples of 1/2% of the total molecule length for $K = 200$. The pair map probability densities can then be expressed in the form of a complete undirected graph, with K nodes corresponding to possible locations, and each edge between node i to j having an edge value equal to the precomputed pair map probability density of $[h_i, h_j]$. A candidate map $(h_1, h_2, \ldots, h_N)$ corresponds to a clique of size N in the graph, and its approximate score corresponds to the smallest edge weight in the clique.

In general, the *clique problem* (for instance, with binary edge weights) is NP-complete and may not result in any asymptotic speedup over the exhaustive search. However, for our problem effective *branch-and-bound* search heuristics can be devised. Consider first the problem of finding just the best clique. We can devise two bounds that can eliminate much of the search space for the best clique:

- The score of any edge of a clique is an upper bound on the score of that clique. If the previous best clique found during a search has a better (higher) score than the score of some edge, all cliques that include this edge can be ruled out.

- For each node in the graph, one can precompute the score of the best edge that includes this node. If the previous best clique found during a search has a better (higher) score than this node score, all cliques that include this node can be ruled out.

As with all branch-and-bound heuristics the effectiveness depends on quickly finding some good solutions, in this case cliques with good scores. We have found that an effective way is to sort all K nodes by the Bayesian scores of the corresponding single cut map. In other words we first try nodes that correspond to restriction site locations that have a high observed cut rate in some orientation of the molecules. Also the nodes corresponding to cut sites of the best overall map so far (with fewer than N cut sites) are tried first.

For data consisting of a few hundred molecules, the branch-and-bound heuristics allows exhaustive search in under a minute on a Sparc System 20 with $N \leq 7$ (with $K = 200$). For $N > 7$, a simple step wise search procedure that searches for the best map (h_1, h_2, ..., h_N) by fixing $N - 7$ nodes based on the previous best map, works well. The $N - 7$ nodes selected are the optimal with respect to a simple metric, for instance, the nodes with the smallest standard error (i.e., ratio of standard deviation to square root of sample size).

Next, the global search is modified to save the best B (typically 8000) cliques of each size and then the exact Bayesian probability density is evaluated at each of these B locations, adding reasonable values for parameters other than (N; h_1, ..., h_N). These parameters can be taken from the the previous best map, or by using some prior values if no previous best map is available. For some best scoring subset (typically 32–64) of these maps gradient search is used to locate the nearest maxima (and also accurate estimates for all parameters), and the best scoring maxima is used as the final estimate for the global maxima for the current value of N.

The branch-and-bound heuristics was modified to find the best B cliques, by maintaining the best B cliques (found so far) in a priority queue (with an ordering based on the approximate score).

4.5.2 Further Improvements

We plan to further investigate several variations to the global search described here:

- For large N the approximate score diverges from the true Bayesian score. To reduce the reliance on the the approximate score the step wise search procedure can be modified to fixing $N-3$ nodes (say) from the previous best map instead of $N-7$. For the same value of B this increases the fraction of the search space that is scored with the exact Bayesian score. Fixing $N - 1$ or even $N - 2$ nodes would allow essentially the entire remaining search space to be scored with the exact Bayesian score. The drawback is that the amount of backtracking has been reduced and hence a wrong cut site found for small N is harder to back out of.

- Instead of searching the space in strictly ascending order of N it is quicker to use a greedy search to locate a good map for a small value of N (say, 5) and then use the more exhaustive search with backtracking to extend it to larger value of N. For large number of cuts (as in BACs) this heuristic leads to significant saving, since the molecule orientations are known (with high probability) once the best map with $N = 5$ is found. With known molecule orientations, even a greedy search using exact Bayesian scores can locate the correct map with high probability. The final more exhaustive search is needed mainly to get a good quality measure for the result.

- To fix the $N - 2$ or $N - 3$ best nodes it might be better to use a greedy search with exact Bayesian scores: Successively try deleting one cut at a time, locating the cut which reduces the exact Bayesian score the least.

4.6 A Quality Measure for the Best Map

As a quality measure for the best map, we use the estimated probability of the dominant mode (peak) of the posterior probability density. This could be computed by integrating the probability density over a small neighborhood of the peak (computed in the parameter space). Our cost function corresponds to a constant multiple of the posterior probability density, as we do not explicitly normalize the cost function by dividing by a denominator corresponding to the integral of the cost over the entire parameter space. To compute the quality measure we make the following simplifying assumption: "All peaks are sharp and the integral of the cost function over a neighborhood where the cost value is larger than a specific amount is proportional to the peak density." Also if we know the N most dominant peaks (typically $N = 64$), we can approximate the integral over all space, by the integral over the N neighborhoods of these peaks. Thus we estimate our quality measure for the best map by the ratio of the value assigned to it (the integral of the cost function in a small neighborhood around it) to the sum of the values assigned to the N best peaks. This, of course, simplifies the computation while producing a rather good estimate. To take into account the sampling errors (when the number of molecules is small) we penalize (reduce) the density of the best map by an estimate of the sampling error. This approach makes the computed quality measure somewhat pessimistic but provides a lower bound.

5 Experimental Results

The following experiments have been conducted with software implementing the *Bayesian Estimation* described in the previous section. In each case, we report the *number of cut sites*, *molecules*, the *quality measure*, the *digest rate* and *cut site standard deviation* reported by the software. The *map error* displays either the RMS error between the map reported by the software and the correct map known by some independent technique (for example complete sequencing if available) in those cases where the software found the right number of cut sites. Otherwise, the software indicates that the map found is *unacceptable*.

5.0.1 Lambda Bacteriophage DNA (I)

Deposited manually using the "peel" technique. Correct map known from sequence data. Data collected: June 1995.

R. Enzyme	Cuts	Mols	Quality	Digest rate	Cut SD	Map Error	
Sca I	6	292	100%	35%	1.82%	0.67%	
Ava I	8	504	99%	32%	1.66%	0.83%	(Fig. 4)

5.0.2 Lambda Bacteriophage DNA (II)

Deposited mechanically (by a robot) as a grid of spots, each spot producing an independent map. Correct map known from sequence data. Data collected: July 1996.

R. Enzyme	Cuts	Mols	Quality	Digest rate	Cut SD	Map Error	
BamH I	5	203	37%	42%	2.82%	1.07%	
BamH I	5	160	100%	45%	2.35%	0.98%	
BamH I	5	257	100%	58%	1.74%	0.79%	
BamH I	5	215	99%	50%	2.61%	0.43%	
BamH I	5	215	100%	61%	1.19%	0.29%	(Fig. 5)
BamH I	7	175	9%	24%	2.25%	Wrong Map	

5.0.3 Human Cosmid Clones

Using a cosmid vector, and deposited as a grid of spots. Map verified by contig and gel electrophoresis as having 6 cuts, with one small fragment ($< 1kB$, and optically undetectable in most of the images) missing [Marked ($*$) in the table below]. Note that the first two rows are the same experiment returning two equally likely answers. Data collected: October 1996.

R. Enzyme	Cuts	Mols	Quality	Digest rate	Cut SD	Map Error	
Mlu I	6	749	50%	38%	2.77%	($*$)	
Mlu I	5	649	50%	31%	2.50%	0.61%	
Mlu I	6	960	100%	50%	2.22%	($*$)	
Mlu I	5	957	72%	26%	2.83%	1.45%	
Mlu I	5	745	99%	37%	2.77%	0.67%	(Fig. 6)
Mlu I	10	852	8%	14%	2.64%	Wrong Map	

6 Conclusion

In this paper, we make three contributions toward the construction of restriction map with optical mapping data.

1. We provide the first detailed model of the data produced by the optical mapping process. We formulate and analyze the worst-case complexity of the problem of constructing restriction map from this data. The model as well as the complexity study has played an important role in the formulation of a Bayesian approach that hinges on the fact that the model is comprehensive and derives its efficiency from the interplay between heuristic global search and exact local search.

2. We formulate a statistical algorithm for this problem that relies on a log-likelihood function derived from a carefully modeled prior distribution. We also derive the update rules for the model parameters and devise an efficient iterative algorithm based on dynamic programming. The multi-modal structure of the prior does not allow a closed-form solution or a local algorithm. This appears consistent with our complexity results showing that the problem is NP-complete. We provide heuristics employing branch-and-bound procedures that bound the search space significantly.

3. We have implemented the algorithm (in C, running on Sparc 20's) and experimented extensively over a period of more than a year. The experiments yield highly accurate maps, consistent with the best result one can expect from the input data.

It may appear that our algorithm is extremely conservative; the detailed modeling as well as the global search may seem to be dispensable specially when one is willing to accept maps that are occasionally wrong and/or relatively frequently inaccurate. We have instead propounded a stronger approach. We justify this on several grounds:

1. The detailed modeling provides a clear physical/statistical interpretation of each step of our algorithm. Should the algorithm ever fail on a set of data, we can immediately trace the source of the error to a specific lack of comprehensiveness of our model and rectify the problem.

2. The approach also allows one to produce not just a single map, but a set of maps ranked by their "quality of goodness." One can then use this information to safeguard the database from being corrupted and provide some very important feedback to the experimenters who could repeat their experiment and gather more data when the estimated qualities are too low.

3. The output of this algorithm is guaranteed to have the optimal accuracy. The demand for this high-accuracy is justified by the fact that even a small loss of accuracy contributes to an exponential growth in the complexity of the "contig" problem and is ultimately a stumbling block to creating genome-wide physical map [GGK+95, Kar93, PW95].

4. Finally, the approach generalizes quite easily to other cases where the data model differs significantly. For instance, with BAC data one can expect the end-fragments to occasionally break and to miss the interior fragments occasionally. Other important situations involve the models for circular (non-linearized) DNA, genomic (uncloned) DNA, data sets consisting of clones of two or more DNA's. Other situation involves augmentation with some more (helpful) data that can be made available by appropriate changes to the chemistry—presence of external standards allowing one to work with absolute fragment sizes, or external labeling disambiguating the orientation or alerting one to the absence of a fragment. The flexibility of our approach derives from its generality and cannot be achieved by the simpler heuristics.

Acknowledgment. Our thanks go to all the people involved in optical mapping and specifically those who formulated and implemented several early heuristics for the restriction map problem. The very first implementation is due to Jason Reed and was facilitated by the image processing algorithms by Ed Huff, Jung-Shih Lo, Davi Geiger and many others. To the research scientists who provided the data and carefully tested the algorithm on these data, we owe a very special thank you: Virginia Clarke, Junping Jing and Joane Edington. Estarose Wolfson worked closely with both computer scientists and the chemists in testing out the software on a vast amount of data. She suffered through the earlier versions of the software and provided many insightful observations.

Our heart-felt thanks go to Brett Porter, Alex Shenker and Ernest Lee who provided the underlying systems support.

Some of the ideas discussed here evolved in weekly group meetings: we thank all the participants of the meeting. We also thank Jitendra Malik, Kurt Mehlhorn, J.R. Murti, Laxmi Parida, Jason Reed, Kurt Riedel, Joel Spencer and Michael Waterman for helpful comments. We wish to thank the anonymous referees, the editor and Ed Huff for detailed comments on the paper.

References

[BSP+90] E. Branscomb, T. Slezak, R. Pae, D. Galas, A.V. Carrano and M.S. Waterman. "Optimizing Restriction Fragment Fingerprinting Methods for Ordering Large Genomic Libraries," *Genomics*, **8**:351–366, 1990.

[BSW84] D. Barker, M. Schafer and R. White. "Restriction Sites Containing CpG Show a Higher Frequency of Polymorphism in Human DNA," *Cell*,**36**:131–138, 1984.

[CAH+95] W. Cai, H. Aburatani, D. Housman, Y. Wang, and D.C. Schwartz. "Ordered Restriction Endonuclease Maps of Yeast Artificial Chromosomes Created by Optical Mapping on Surfaces," *Proc. Natl. Acad. Sci., USA*, **92**:5164–5168, 1995.

[CJI+96] W. Cai, J. Jing, B. Irvine, L. Ohler, E. Rose, U. Kim, Shizuya, M. Simon, T. Anantharaman, B. Mishra and D.C. Schwartz. "High Resolution Restriction Maps of Bacterial Artificial Chromosomes Constructed by Optical Mapping," Submitted to *Genomics*, 1997.

[CVS95] W. Cedeno, V.R. Vemuri and T. Slezak. "Multiniche Crowding in Genetic Algorithms and Its Application to the Assembly of DNA Restriction-Fragments," *Evolutionary Computation*, **2**(4):321–345, 1995.

[CW92] G.A. Churchill and M.S. Waterman. "The Accuracy of DNA Sequences: Estimating Sequence Quality," *Genomics*, **14**:89–98, 1992.

[Chu97] G. Churchill. "Monte Carlo Sequence Alignment," In *Proceedings First Annual Conference on Computational Molecular Biology*, (RECOMB97), ACM Press, 93–97, 1997.

[DHM97] V. Dančík, S. Hannenhalli and S. Muthukrishnan. "Hardness of Flip-Cut Problems Optical Mapping," Unpublished Manuscript, 1997.

[DLR77] A.P. Dempster, N.M. Laird and D.B. Rubin. "Maximum Likelihood from Incomplete Data via the EM Algorithm," *J. Roy. Stat. Soc.*, **39**(1):1–38, 1977.

[GJ79] M.R. Garey and D.S. Johnson. *Computer and Intractability: A Guide to the Theory of NP-Completeness*, W.H. Freeman and Co., San Fransisco 1979.

[GGK+95] P.W. Goldberg, M.C. Golumbic, H. Kaplan and R. Shamir. "Four Strikes Against Physical Mapping of DNA," *J. Comp. Bio.*, **2**(1):139–152, 1995.

[GM87] L. Goldstein and M.S. Waterman. "Mapping DNA by Stochastic Relaxation," *Adv. Appl. Math.*, **8**:194–207, 1987.

[GM93] U. Grenander and M.I. Miller. "Representations of Knowledge in Complex Systems (with discussions)," *J. Roy.Stat. Soc. B.*, **56**:549–603, 1993.

[HW92] X. Huang and M.S. Waterman. "Dynamic Programming Algorithms for Restriction Map Comparison," *Comp. Appl. Bio. Sci.*, **8**:511–520, 1992.

[HRL+95] E.J. Huff, J. Reed, I. Lisanskiy, J.-S. Lo, B. Porter, T. Anantharaman, B. Mishra, D. Geiger and D.C. Schwartz. "Automatic Image Analysis for Optical Mapping," In *1995 Genome Mapping and Sequencing Conference*, Cold Spring Harbor, New York, May 10–14, 1995.

[JRH+96] J. Jing, J. Reed, J. Huang, X. Hu, V. Clarke, D. Housman, T. Anantharaman, E. Huff, B. Mishra, B. Porter, A. Shenker, E. Wolfson, C. Hiort, R. Cantor and D.C. Schwartz. "Automated High Resolution Optical Mapping Using Arrayed, Fluid Fixed DNA Molecules," Submitted to *Science*, 1996.

[Kar93] R.M. Karp. "Mapping the Genome: Some Combinatorial Problems arising in Molecular Biology," In *Proc. of 25th Ann. ACM Symp. on the Theory of Computing*, 278–285, 1993.

[KH92] D.J. Kevles and L. Hood (Editors). *The Code of Codes*. Harvard University Press, Ma, 1992.

[Kra88] M. Krawczak. "Algorithms for Restriction-Site Mapping of DNA Molecules," In *Proc. Natl. Acad. Sciences USA*, **85**:7298–7301, 1988

[Lan95a] E.S. Lander. "Mapping Heredity: Using Probabilistic Models and Algorithms to Map Genes and Genomes," *Notices of the AMS*, **42**(7):747–753, 1995. Adapted from "Calculating the Secrets of Life," National Academy of Sciences, 1995.

[Lan95b] E.S. Lander. "Mapping Heredity: Using Probabilistic Models and Algorithms to Map Genes and Genomes (Part II)," *Notices of the AMS*, **42**(8):854–858, 1995. Adapted from "Calculating the Secrets of Life," National Academy of Sciences, 1995.

[LW88] E.S. Lander and M.S. Waterman. "Genomic Mapping by Fingerprinting Random Clones: A Mathematical Analysis," *Genomics*, **2**:231–239, 1988.

[MBC+95] X. Meng, K. Benson, K. Chada, E. Huff and D.C. Schwartz. "Optical Mapping of Lambda Bacteriophage Clones Using Restriction Endonuclease," *Nature Genetics*, **9**:432–438, 1995.

[MP96] S. Muthukrishnan and L. Parida. "Towards Constructing Physical Maps by Optical Mapping: An Effective Simple Combinatorial Approach." In *Proceedings First Annual Conference on Computational Molecular Biology*, (RECOMB97), ACM Press, 209–215, 1997.

[Nic94] D.S.T. Nicholl. *An Introduction to Genetic Engineering*, Cambridge University Press, 1994.

[Pev90] P.A. Pevzner. "DNA Physical Mapping." *Computer Analysis of Genetic Texts*, 154–158, 1990

[PW95] P.A. Pevzner and M.S. Waterman. "Open Combinatorial Problems in Computational Molecular Biology," In *Proc. of the 3rd. Israel Symp. on Theory of Computing and Systems*, January, 1995.

[Pri95] S.B. Primrose. *Principles of Genomic Analysis: A Guide to Mapping and Sequencing DNA from Different Organisms*, Blackwell Science Ltd., Oxford, 1995.

[Ree97] J. Reed. *Optical Mapping*, Ph. D. Thesis, NYU, June 1997 (Expected).

[SCH+95] A. Samad, W.W. Cai, X. Hu, B. Irvin, J. Jing, J. Reed, X. Meng, J. Huang, E. Huff, B. Porter, A. Shenker, T. Anantharaman, B. Mishra, V. Clarke, E. Dimalanta, J. Edington, C. Hiort, R. Rabbah, J. Skiadas, and D.C. Schwartz. "Mapping the Genome One Molecule At a Time—Optical Mapping," *Nature*, **378**:516–517, 1995.

[SLH+93] D.C. Schwartz, X. Li, L.I. Hernandez, S.P. Ramnarain, E.J. Huff and Y.K. Wang. "Ordered Restriction Maps of *Saccharomyces cerevisiae* Chromosomes Constructed by Optical Mapping," *Sciences*, **262**:110–114, 1993.

[WHS95] Y.K. Wang, E.J. Huff and D.C. Schwartz. "Optical Mapping of the Site-directed Cleavages on Single DNA Molecules by the RecA-assisted Restriction Endonuclease Technique," In *Proc. Natl. Acad. Sci. USA*, **92**:165–169, 1995.

[WSK84] M.S. Waterman, T.F. Smith and H. Katcher. "Algorithms for Restriction Map Comparisons," *Nucleic Acids Research*, **12**: 237–242, 1984.

[Wat89] M.S. Waterman (Editor). *Mathematical Methods for DNA Sequences*, CRC Press, Florida, 1989.

[Wat95] M.S. Waterman. *An Introduction to Computational Biology: Maps, Sequences and Genomes*, Chapman Hall, 1995.

[Wat77] J. Watson. *Molecular Biology of the Gene*, W.A. Benjamin, Inc., Ma., 1977.

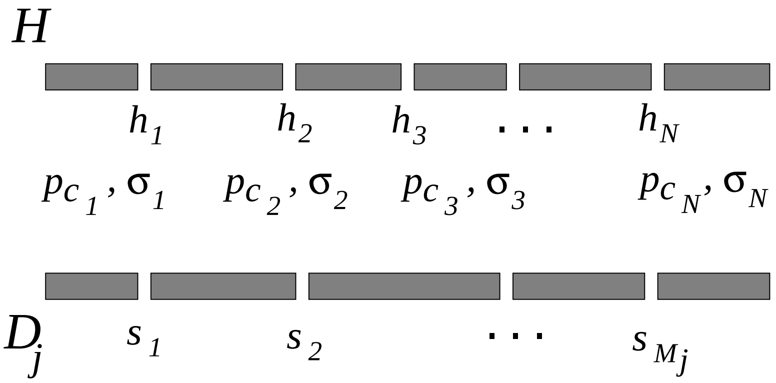

Figure 1: A statistical model of the cuts

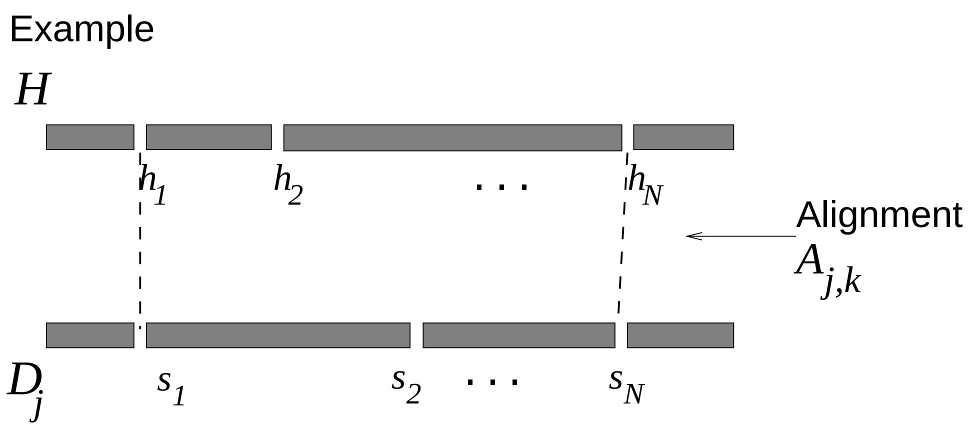

Figure 2: An example of the alignment detection

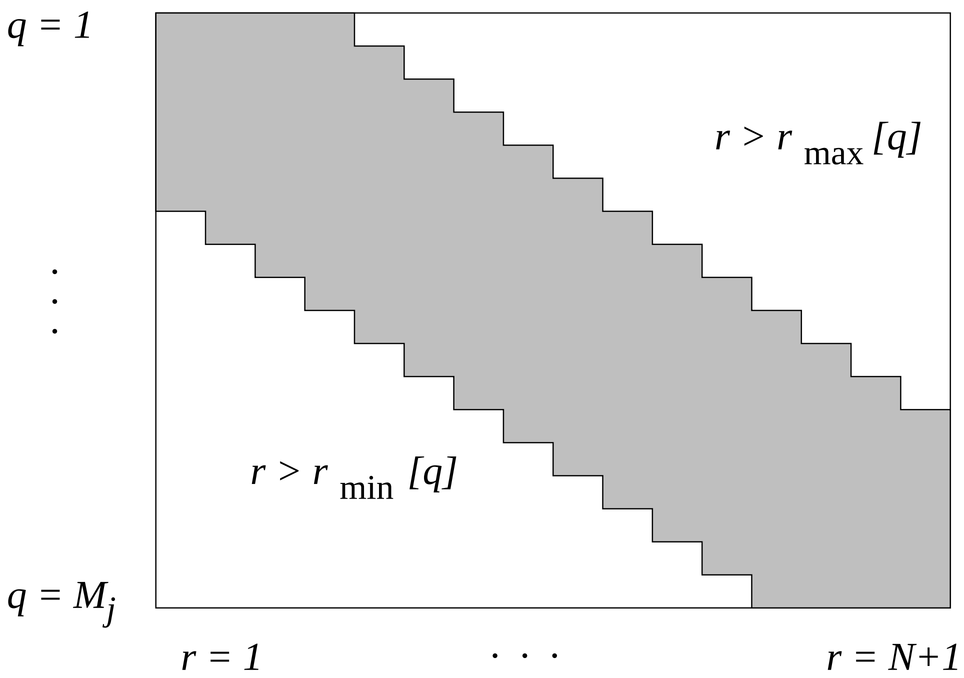

Figure 3: A variable block diagonal matrix for the dynamic programming

```
molecules=504,cuts=1441,uncut molecules=39,best 3 maps:

map1:cuts=8,P=99.5354%,good mols=79.41%,digest rate=0.3251,false cuts=0.3315,SD=0.0166
 9 frags: 0.09507 0.31530 0.03891 0.13294 0.07145 0.04462 0.08265 0.03984 0.17923
 8 cuts : 0.09507 0.41037 0.44928 0.58221 0.65366 0.69828 0.78093 0.82077
cut SDs : 0.01423 0.01653 0.01663 0.01786 0.01689 0.01823 0.01782 0.01482
 counts :   152.4   118.9   116.9   133.0   137.4   123.1   112.9   146.1

map2:cuts=9,P= 0.4646%,good mols=79.99%,digest rate=0.2860,false cuts=0.3458,SD=0.0150
10 frags: 0.09503 0.31652 0.03876 0.12911 0.05924 0.03397 0.03859 0.07162 0.03849 0.17867
 9 cuts : 0.09503 0.41155 0.45031 0.57942 0.63866 0.67263 0.71122 0.78284 0.82133
cut SDs : 0.01319 0.01488 0.01513 0.01593 0.01658 0.01360 0.01690 0.01563 0.01370
 counts :   147.3   115.8   110.3   122.8    93.1   107.7    89.9   110.5   140.6

map3:cuts=10,P= 0.0000%,good mols=75.70%,digest rate=0.2596,false cuts=0.3445,SD=0.0133
11 frags: 0.09489 0.31828 0.03800 0.11818 0.02946 0.04939 0.02851 0.03574 0.07067 0.03786 0.17902
10 cuts : 0.09489 0.41317 0.45117 0.56935 0.59881 0.64820 0.67671 0.71245 0.78312 0.82098
cut SDs : 0.01198 0.01335 0.01366 0.01294 0.01406 0.01344 0.01261 0.01549 0.01407 0.01245
 counts :   136.5   108.3    99.2    78.6    74.3    88.1    93.1    77.9   101.9   132.7

RMS Map Error=0.00826 (relative to map1)
```

Figure 4: Map computed using the Bayesian approach. Correct ordered restriction map (from sequence data) for the Lambda Bacteriophage DNA (I) with Ava I is: (0.09732, 0.39992, 0.43295, 0.57497, 0.65187, 0.69065, 0.78789, 0.82240)

```
molecules=215,cuts=523,uncut molecules=34,best 3 maps:

map1:cuts=5,P=100.0000%,good mols=60.4%,digest rate=61.4%,false cuts=0.14,SD=0.0119
 6 frags: 0.14016 0.14444 0.13700 0.11331 0.35012 0.11496
 5 cuts : 0.14016 0.28461 0.42161 0.53492 0.88504
cut SDs : 0.01166 0.01110 0.01180 0.01245 0.01232
 counts :    62.0    75.3    92.2    86.6    59.9

map2:cuts=6,P= 0.0000%,good mols=60.0%,digest rate=48.0%,false cuts=0.13,SD=0.0118
 7 frags: 0.13921 0.14449 0.13165 0.01029 0.10845 0.35102 0.11489
 6 cuts : 0.13921 0.28370 0.41536 0.42564 0.53409 0.88511
cut SDs : 0.01155 0.01102 0.01137 0.01188 0.01261 0.01207
 counts :    58.8    70.1    45.0    47.1    79.6    56.9

map3:cuts=6,P= 0.0000%,good mols=60.0%,digest rate=48.0%,false cuts=0.15,SD=0.0113
 7 frags: 0.13932 0.14426 0.13718 0.10475 0.01688 0.34333 0.11428
 6 cuts : 0.13932 0.28358 0.42076 0.52551 0.54239 0.88572
cut SDs : 0.01138 0.01072 0.01157 0.01150 0.01108 0.01180
 counts :    58.9    70.3    85.2    42.1    44.8    56.0

RMS Map Error=0.00287 (relative to map1)
```

Figure 5: Map computed using the Bayesian approach. Correct ordered restriction map (from sequence data) for the Lambda Bacteriophage DNA (II) with BamH I is: (0.13960, 0.28870, 0.42330, 0.53930, 0.88650)

```
molecules=745,cuts=1755,uncut molecules=66,best 3 maps:

map1:cuts=5,P=99.8579%,good mols=80.25%,digest rate=0.3696,false cuts=0.5325,SD=0.0277
 6 frags: 0.21919 0.26956 0.09810 0.09723 0.18784 0.12808
 5 cuts : 0.21919 0.48875 0.58685 0.68408 0.87192
cut SDs : 0.02782 0.03030 0.02472 0.02897 0.02748
 counts :   210.3   194.2   278.6   218.6   203.1

map2:cuts=6,P= 0.1421%,good mols=77.15%,digest rate=0.3064,false cuts=0.5343,SD=0.0250
 7 frags: 0.21963 0.26424 0.08223 0.03746 0.08386 0.18410 0.12848
 6 cuts : 0.21963 0.48387 0.56610 0.60356 0.68742 0.87152
cut SDs : 0.02525 0.02600 0.02454 0.02355 0.02561 0.02487
 counts :   188.1   160.7   167.6   173.2   185.9   181.1

map3:cuts=7,P= 0.0000%,good mols=90.61%,digest rate=0.2513,false cuts=0.5985,SD=0.0241
 8 frags: 0.21327 0.27029 0.08288 0.03357 0.07312 0.06696 0.13289 0.12702
 7 cuts : 0.21327 0.48356 0.56644 0.60002 0.67314 0.74009 0.87298
cut SDs : 0.02475 0.02501 0.02392 0.02221 0.02226 0.02620 0.02468
 counts :   183.0   166.4   169.0   176.4   175.9   125.7   191.2

RMS Map Error=0.00665 (relative to map1)
```

Figure 6: Map computed using the Bayesian approach. Correct fingerprint (from gel electrophoresis) for the Human Cosmid Clone with Mlu I is: (0.09362, 0.09974, 0.12643, 0.19763, 0.21862, 0.26396). This fingerprint omits one small ($< 1kB$) fragment. The correct ordered restriction map consistent with the fingerprint data is: (0.21862, 0.48258, 0.57620, 0.67594, 0.87357).