Fast and Cheap Genome wide Haplotype Construction via Optical Mapping
NYU Technical Report #TR2004-852

Thomas Anantharaman[1]*, Venkatesh Mysore[2], and Bud Mishra[2,3]

[1] Wisconsin Biotech Center, University of Wisconsin, 425 Henry Mall, Madison WI, USA 53706.

[2] Courant Institute of Mathematical Sciences, New York University, 251 Mercer Street, New York, NY, USA 10012.

[3] Cold Spring Harbor Lab, 1 Bungtown Road, Cold Spring Harbor, NY, USA 11724.

June 7, 2004

ABSTRACT **We describe an efficient algorithm to construct genome wide haplotype restriction maps of an individual by aligning single molecule DNA fragments collected with Optical Mapping technology. Using this algorithm and small amount of genomic material, we can construct the parental haplotypes for each diploid chromosome for any individual, one from the father and the other from the mother. Since such haplotype maps reveal the polymorphisms due to single nucleotide differences (SNPs) and small insertions and deletions (RFLPs), they are useful in association studies, studies involving genomic instabilities in cancer, and genetics. For instance, such haplotype restriction maps of individuals in a population can be used in association studies to locate genes responsible for genetics diseases with relatively low cost and high throughput.**

If the underlying problem is formulated as a combinatorial optimization problem, it can be shown to be NP-complete (a special case of K-population problem). But by effectively exploiting the structure of the underlying error processes and using a novel analog of the Baum-Welch algorithm for HMM models, we devise a probabilistic algorithm with a time complexity that is linear in the number of markers.

The algorithms were tested by constructing the first genome wide haplotype restriction map of the microbe *T. Pseudoana*, as well as constructing a haplotype restriction map of a 120 Megabase region of Human chromosome 4. The frequency of false positives and false negatives was estimated using simulated data. The empirical results were found very promising.

*To whom correspondence should be addressed. E-mail: tsa@biostat.wisc.edu, Tel: 608-3470637

1 Introduction

Diploid organisms, such as humans, carry two mostly similar copies of each chromosome, referred to as haplotypes. Variations in a large population of haplotypes at specific loci are called polymorphisms. The co-associations of these variations across the loci indices are of intense interest in disease research.

The polymorphic marker of most interest has been the Single Nucleotide Polymorphism (SNP). There are estimated to be 15 million such markers over the entire human genome (including only SNPs where the minor allele occurs at least 5% of the time) and a realistic genome wide association study would require at least 1000 samples to be tested for each of these 15 million SNPs. By taking advantage of linkage disequilibrium [1, 7, 8, 17] the number of SNPs that need to be tested can be reduced to about 300,000. The main limitation of most SNP based approaches is that each SNP is assayed separately and hence it is not possible to accurately infer the exact haplotype map for a particular individual (which we will refer to as an *individual haplotype map* and is sometimes called *one-individual haplotype map* or *personal haplotype map*) from SNP assays alone. Instead, the phase is inferred statistically from a large population of SNP data. These phasing algorithms use assumptions such as parsimony in the total number of different haplotypes in the population, the Hardy-Weinberg equilibrium, perfect phylogeny to combinatorially constrain the possible haplotypes [5, 9, 10, 12, 14] or alternatively, use statistical approaches [19, 21] (available as PHASE and HAPLOTYPER software packages). The statistical algorithms have tended to be bit more accurate, but unforgivingly slow. The results of all these algorithms are

referred to as population haplotype maps and are typically only 90–95% accurate in any particular map on average and would thus rarely be correct in any individual haplotype. The inaccuracies emanate from multiple, complex and unwieldy sources of errors: population stratifications, violation to perfect phylogeny assumption (with hemizygous deletions or gene conversion), or corrupted data manifesting as genotype errors. Thus, it is necessary that we explore new direct and cost-effective methods, where a single individual's genome is analyzed to create the haplotype sequences or maps without making any overly generalized assumptions, using population wide statistics, or requiring the availability of parental genomes (as in trio-studies).

For a genotyping method to be able to correctly determine the phasing between neighboring polymorphic markers in every individual haplotype map, it must ultimately be able to test single DNA fragments containing 2 or more heterozygous polymorphic markers in a single test. It is possible, of course, to assemble individual haplotype maps by sequencing the individual's entire genome using a modified sequence assembly algorithm [16, 24] but the cost of doing this is prohibitive[1].

Here, we propose a direct and more cost-effective approach using the fairly well developed single molecule technology of Optical Mapping. We demonstrate the feasibility of constructing such an accurate individual haplotype map of restriction sites through pilot-scale experiments and simulation studies. A brief description of the data generated by Optical Mapping technology is given in the following section, and many more details of the technology can be found in the literature (e.g. [15, 18, 25]). Unfortunately, the input genomic data that can be collected from a single DNA molecule by the best chemical and optical methods (such as those used in Optical Mapping) are corrupted by many poorly understood noise processes. Thus to make this system feasible, the biggest challenge has been in developing accurate Bayesian probabilistic models of errors for experiment design and efficient maximum likelihood algorithms to achieve accuracy with sufficiently redundant data.

Each individual haplotype map of restriction sites will only detect a small fraction of all polymorphisms in the human genome, but using the same linkage disequilibrium assumption mentioned previously, approximately 8 individual haplotype restriction maps will contain more than the 300,000 SNPs required to infer all other known polymophisms in the individual genome. Even with 50 fold data redundancy required, all date required for 8 individual haplotype restriction maps can be collected for under $1000.

In this paper, we present an unambiguous mathematical formulation of the problem (section 2), combinatorial complexity of the resulting problem (also section 2), an efficient probabilistic algorithmic solution built upon a detailed Bayesian model of the underlying error sources (section 3) and finally, its complexity analysis (section 4). We also provide data on the performance of the algorithm on real and simulated examples (section 5) and conclude with a discussion of the future problems (section 6).

[1]This cost has been estimated to be over $10 million per individual

2 Problem Formulation

Our problem can be formulated mathematically as follows: We assume that all individual single molecule DNA fragments are derived from a diploid genome (ignoring the case of sex chromosomes) with two copies of homologous chromosomes. Each DNA fragment is further mapped by cleavage with a restriction enzyme of choice and imaged by an imaging algorithm to produce an ordered sequence of "restriction fragment lengths" or equivalently, "restriction sites." The variations in these restriction fragment lengths are primarily due to RFLPs as well as SNPs at the restriction sites. Additionally, there are further variations introduced by the experimental process and could be assumed due to: sizing errors, partial digestion, short missing restriction fragments, false cuts, ambiguities in the orientation, optical chimerisms, etc. Thus, the genomes may be represented as two haplotype restriction maps, H_1 and H_2, for the same individual which differ only slightly from a genotype restriction map H by a small number of short insertions, deletions and SNPs that coincide with restriction sites. All such maps, H, H_1 and H_2, are assumed to be representable as a sequence of restriction sites (e.g. $H_{2,i}$, with indices $0 \leq i \leq (N+1)$, where $H_{2,0}$ and $H_{2,N+1}$ represent the chromosome ends), but are unknown. However, short DNA fragments of around 500 Kb derived from such maps, and further corrupted by experimental noise processes can be readily generated at high throughput and very low cost using a technology like Optical Mapping [15, 18, 25]. These short DNA fragments will be written as D_k, with indices $1 \leq k \leq M$, where M is the number of data fragments and each data fragment is in turn represented as a sequence of restriction sites (e.g. $D_{k,j}$, $0 \leq j \leq m_k + 1$) and can be aligned globally to create an estimate of genotype map H using algorithms described previously [3].

The algorithmic problem, we wish to study, is to further separate H into two maps H_1 and H_2 in such a manner that each data fragment D_k is aligned well to one haplotype or other and that H_1 and H_2 differ from H only by modifications consistent with SNPs or RFLPs polymorphisms.

Thus, ultimately, this problem corresponds to a problem of

refining a multiple map alignment in to two families, starting with one global alignment. A combinatorial generalization, where the number of such families is arbitrarily large ($k > 1$) and the cost of each alignment is arbitrarily unconstrained, has been shown to lead to computationally infeasible problems. See [20] for the proof of NP-completeness as well as a probabilistic analysis to show conditions under which the problem can be solved efficiently with a probability close to one. The key to an effective solution of these problems relies on careful experiment design (e.g., choice of coverage, restriction enzyme, experimental conditions, etc.) to ensure conditions under which a polynomial time probabilistic algorithm will work with high probability in conjunction with a Bayesian error model that encodes the error processes properly.

To construct individual haplotype maps from Optical Mapping data we use a mixture hypothesis of pairs of maps H_1 and H_2 for each chromosome, corresponding to the correct restriction map of the two parental chromosomes. We first assemble the data into a regular map of the entire genome and use this assembly to separate the data into distinct chromosome sets: all maps from the same chromosome belonging to a pair will be included in the same set. We then use a probabilistic model of the errors in the data to derive conditional probability density expressions $f(D_k|H_1)$ and $f(D_k|H_2)$, and apply Bayes rule to maximize a score for the best alignment with respect to proposed H_1 and H_2, Equation 1.:

$$\begin{aligned} & f(H_1, H_2|D_1, \ldots, D_M) \qquad (1) \\ & \quad \propto \quad f(H_1, H_2) f(D_1, \ldots, D_M|H_1, H_2) \end{aligned}$$

The first term on the right side is the prior probability of H_1 and H_2 and we just use a low prior probability for each polymorphism (difference in H_1 vs. H_2). For the conditional probability term, we can assume each map is a statistically independent sample from the genome and that the mapping errors are drawn from i.i.d. distributions and hence write:

$$f(D_1, \ldots, D_M|H_1, H_2) = \prod_{k=1}^{M} \frac{[f(D_k|H_1) + f(D_k|H_2)]}{2} \quad (2)$$

The conditional terms of the form $f(D_k|H_i)$ above can be written as a summation over all possible (mutually exclusive) alignments between the particular D_k and H_i, and for each alignment the probability density is based on an enumeration of the map errors in the alignment and multiplying together the probability associated with each error under some suitable error model. The exact form of the error models suitable for Optical Mapping is described in the next section, but for almost any error models used the sum of the probability for all alignments can be computed effectively using dynamic programming. In the next two sections, we will derive the dynamic programming recurrence relations for this problem and show how to implement the algorithm with attractive computational complexities. Using these alignment algorithms, we will see how it is possible to quickly search through the space of all haplotype pairs to find the most plausible ones consistent with the data.

Other methods for assembling Optical Mapping data for relatively short clones into genotype restriction maps exist, e.g., ones based on Markov Chain Monte Carlo search [22] or Maximum Likelihood [23] or heuristic scoring functions [13]. However these methods are yet to be scaled to map whole genomes. Even for restriction maps of BAC clones these methods produce the correct map only 50% of the time [13] and have no way of signaling when the method fails. Bayesian algorithms [3, 4], similar to the ones described here, are routinely used by untrained chemists and biologists to assemble genotype restriction maps with a success rate of higher than 90%, and produce p-values to signal failure rather than produce an incorrect map.

3 Algorithm

3.1 Computing Conditional Probabilities for a Hypothesis

Theorem 3.1 *Consider an arbitrary alignment between the data D and the hypothesis H, J^{th} restriction site of D matching the I^{th} restriction site of H. We will denote this aligned pair by $J \mapsto I$.*

Let the probability density of the unaligned portion on the left and right end of such an alignment be denoted by $f_{ur}(I, J)$ on the right end if $J \mapsto I$ is the rightmost aligned pair, and $f_{ul}(I, J)$ on the left end if $J \mapsto I$ is the leftmost aligned pair.

In addition, the following probability density functions f_m and f_a denote the following:

$$\begin{aligned} f_m(I, P) &= Pr[H[I..P] \text{ is missing in the observed data } D]. \\ f_a(I, J, P, Q) &= Pr[H[I..P] \text{ is an aligned region but not a missing fragment with respect to the observed data region } D[J..Q]]. \end{aligned}$$

We assume that $I < P$ and $J < Q$.

Then the following holds:

$$f(D|H) = \sum_{I=1}^{N} \sum_{J=0}^{m+1} f_{ul}(I, J) f(D[J..m+1]|H[I..N] \wedge J \mapsto I).$$

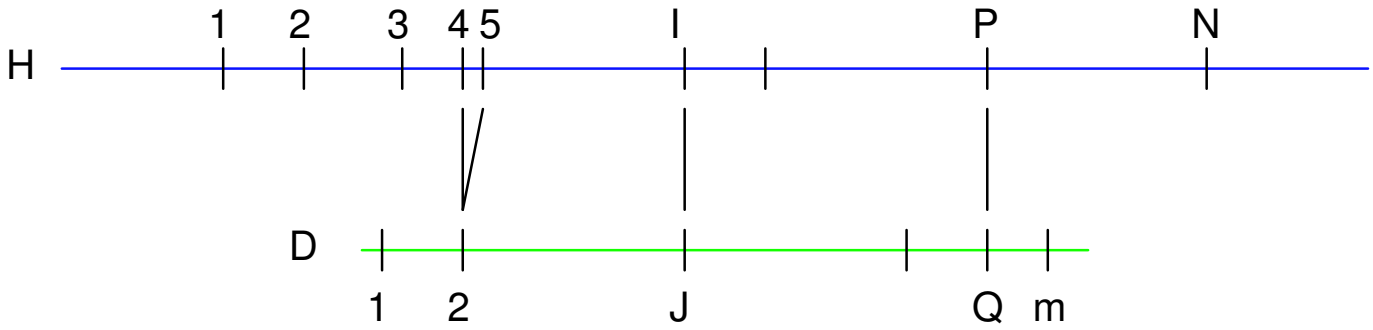

Figure 1: To define the notation required we consider a single arbitrary alignment between a particular data D and hypothesis H. Recall that N is the number of restriction sites in H and m the number of restriction sites in D. Any arbitrary alignment between D and H can be described as a list of pairs of restriction sites from H and D that describes which restriction site from H is aligned with which restriction site from D. As an example, Here the alignment consists of 4 aligned pairs $(4,2)$, $(5,2)$, (I,J) and (P,Q). Notice that not all restriction sites in H or D need be aligned. For example between aligned pairs (I,J) and (P,Q) there is one misaligned site on H and D each, corresponding to a missing site (false-negative) and extra-site (false-positive) in D. In this alignment a true small fragment between sites 4 and 5 in H are missing from D, which is shown by aligning both sites 4 and 5 in H with the same site 2 in D. Note that if two or more consecutive fragments in H are all missing in D, this would be described by aligning all sites for the missing fragments in H with the same site in D (rather than showing only the outermost of this set of consecutive sites in H aligned with D, for example). The expression for the conditional probability density of any alignment, such as the one here, can be written as the product of a number of probability terms corresponding to the regions of alignment between each pair of aligned sites, plus one probability term for each unaligned region at the two ends of the alignment.

$$
\begin{aligned}
& f(D[J..m+1]|H[I..N] \wedge J \mapsto I) \\
&= f_{ur}(I,J) \\
&\quad + f_m(I,I+1) f(D[J..m+1]|H[I+1..N] \wedge J \mapsto (I+1)) \\
&\quad + \sum_{P=I+1}^{N} \sum_{Q=J+1}^{m+1} \\
&\qquad f_a(I,J,P,Q) f([Q..m+1]|H[P..N] \wedge Q \mapsto P)
\end{aligned}
$$

In particular, if the intermediate values are kept in a DP table $\mathsf{A}_{\mathrm{suf}}[I,J]$

$$\mathsf{A}_{\mathrm{suf}}[I,J] = f(D[J..m+1]|H[I..N] \wedge J \mapsto I)$$

then it is easily seen that $f(D|H)$ *can be computed exactly in* $O(m^2N^2)$ *time and* $O(mN)$ *space, assuming that* f_m *and* f_a *are* $O(1)$ *time functions and* f_{ul} *and* f_{ur} *are* $O(N)$ *time functions.* □

In a later section we will see how to reduce the complexity to linear time when we only require an ϵ-approximate value $\widetilde{f}$

$$f(D|H) - \epsilon < \widetilde{f}(D|H) < f(D|H) - \epsilon,$$

for the probability density function arising in the context of optical mapping as follows:

$$
\begin{aligned}
f_m(I,I+1) &= P_\nu^{H_{I+1}-H_I} \\
f_a(I,J,P,Q) &= \lambda^{Q-J-1} P_d (1-P_d)^{P-I-1} (1-P_\nu)^{H_P-H_I} \\
&\quad \times G_{(H_P-H_I),\sigma^2(H_P-H_I)}(D_Q - D_J),
\end{aligned}
$$

where

$$\begin{aligned} P_d &= \text{the digest rate,} \\ \lambda &= \text{the false-positive site rate,} \\ \sigma^2 h &= \text{the Gaussian sizing error variance for a fragment of size } h, \\ P_\nu &= \text{the probability of missing a fragment of unit size, and} \\ R_e &= \text{the breakage rate of DNA (the inverse of the expected fragment size).} \end{aligned}$$

For a random variable x following a Gaussian distribution $\mathcal{N}(\mu, \sigma^2)$, the probability density value at d is

$$G_{\mu,\sigma^2}(d) = \frac{\exp[-(d-\mu)^2/2\sigma^2]}{\sqrt{2\pi}\sigma}.$$

The exact form of the functions for f_{ul} and f_{ur} for Optical Mapping are complicated, but not very important in understanding the complexity of the algorithm; thus a detailed discussion is omitted here, but can be seen in the appendix. Note that, at first glance, f_{ul} and f_{ur} may appear to be $O(N)$ time functions, but it is easily seen that they permit $O(1)$ ϵ-approximation.

As it has been shown elsewhere [2], a good approximate location of the best alignment between D and H can be determined in $O(1)$ expected time, if the conditional probability density has been previously evaluated for a similar H or alternatively, through a geometric hashing algorithms. Only a $O(1)$-width band of the DP table needs to be evaluated to compute an ϵ-approximation $\widetilde{f}(D|H)$. In particular, the band width of the DP table used in practice is usually about $\Delta = 8$; more generally for Optical mapping Δ is bounded by

$$(1-P_d)^{\Delta-1} = \epsilon, \quad \text{or} \quad \Delta = 1 + \frac{\ln(\epsilon)}{\ln(1-P_d)}.$$

With this approach we achieve a reduced time complexity of $O(\min(m, N))$ (more explicitly, $O(\min(m\Delta^3, N))$).

3.2 Recomputing Conditional Probabilities for a Modification to Hypothesis

We next consider following problem: How can one re-evaluate the conditional probability distribution function, $f(D|H' = p(H))$ when the new hypothesis, H', has been obtained by locally changing H in just one place (corresponding to a polymorphism). There are three cases to consider. We study one of the three cases here in detail and refer the reader to the appendix: for details. The omitted cases are similar but tedious.

We may obtain H' by

1. Deleting one of the existing restriction sites in H, as the site may contain a heterozygous SNP;

2. Adding a new restriction site at a specified location in H, symmetrical to the previous case;

3. Increasing or decreasing a restriction fragment length in H, an RFLP;

Consequently, we may also need to compute the first and second derivative of $f(D|H)$ relative to the change in any fragment size in H.

Theorem 3.2 *Consider an arbitrary alignment between the data D and the hypothesis H, J^{th} restriction site of D matching the I^{th} restriction site of H. Using the notations of the previous subsection, we write:*

$$\begin{aligned} \mathsf{A}_{\text{suf}}[I,J] &= f(D[J..m+1]|H[I..N] \wedge J \mapsto I), \text{ and} \\ \mathsf{A}_{\text{pref}}[I,J] &= f(D[0..J]|H[1..I] \wedge J \mapsto I). \end{aligned}$$

Then

$$\begin{aligned} &\mathsf{A}_{\text{suf}}[I,J] \\ &= f_{ur}(I,J) + f_m(I,I+1)\mathsf{A}_{\text{suf}}[I+1,J] \\ &\quad + \sum_{P=I+1}^{N} \sum_{Q=J+1}^{m+1} f_a(I,J,P,Q)\mathsf{A}_{\text{suf}}[P,Q], \end{aligned}$$

and similarly,

$$\begin{aligned} &\mathsf{A}_{\text{pref}}[I,J] \\ &= f_{ul}(I,J) + \mathsf{A}_{\text{pref}}[I-1,J]f_m(I-1,I) \\ &\quad + \sum_{P=1}^{I-1} \sum_{Q=0}^{J-1} \mathsf{A}_{\text{pref}}[P,Q]f_a(I,J,P,Q), \end{aligned}$$

If $H \setminus \{H_K\}$ is obtained from H by deleting the site H_K,

then

$$
\begin{aligned}
& f(D|H \setminus \{H_K\}) \\
&= Pr[\textit{Alignments with rightmost aligned } I < K] \\
&\quad + Pr[\textit{Alignments with leftmost aligned } J > K] \\
&\quad + Pr[\textit{Alignments with a fragment spanning } H[K-1..K+1]] \\
&= \sum_{I=1}^{K-1} \sum_{J=0}^{m+1} \mathsf{A}_{\text{pref}}[I,J] f_{ur}^{(-H_k)}(I,J) \\
&\quad + \sum_{I=K+1}^{N} \sum_{J=0}^{m+1} f_{ul}^{(-H_k)}(I,J) \mathsf{A}_{\text{suf}}[I,J] \\
&\quad + \mathbb{I}_{K<N} \sum_{J=0}^{m+1} \mathsf{A}_{\text{pref}}[K-1,J] f_m(K-1,K+1) \mathsf{A}_{\text{suf}}[K+1,J] \\
&\quad + \sum_{J=0}^{m+1} \sum_{I=1}^{K-1} \sum_{P=K+1}^{N} \sum_{Q=J+1}^{m+1} \mathsf{A}_{\text{pref}}[I,J] \frac{f_a(I,J,P,Q)}{1-P_d} \mathsf{A}_{\text{suf}}[P,Q],
\end{aligned}
$$

where $f_{ul}^{(-H_k)}$ and $f_{ur}^{(-H_k)}$ are computed respectively from f_{ul} and f_{ur} by suitable simple modifications.

Then it is seen that $f(D|H \setminus \{H_K\})$, $\forall K\ 1 \le K \le N$, can be computed exactly in $O(m^2N^2)$ time and $O(mN)$ space, assuming that f_m and f_a are $O(1)$ time functions and f_{ul} and f_{ur} are $O(N)$ time functions.

Proof Sketch:

We omit the derivation of the recurrence relation. In order to see how the stated computational complexity can be achieved, observe that: (1) We can amortize the cost of computing $f(D|H \setminus \{H_K\})$ over all $1 \le K \le N$ by evaluating each term with $f_a(I,J,P,Q)$, $f_{ur}^{(-H_k)}(I,J)$, $f_{ul}^{(-H_k)}(I,J)$ just once. For example, any term

$$\mathsf{A}_{\text{pref}}[I,J] \frac{f_a(I,J,P,Q)}{1-P_d} \mathsf{A}_{\text{suf}}[P,Q]$$

will be present in the summation of $f(D|H \setminus \{H_K\})$ for all $I < K < P$ and absent for all other K.

(2) If all $f(D|H \setminus \{H_K\})$ for all $1 \le K \le N$ are summed into a table $F[K = 1..N]$ where each

$$F[K] = f(D|H \setminus \{H_K\}) - f(D|H \setminus \{H_{K-1}\})$$

except $F[1] = f(D|H \setminus \{H_1\}))$, we just need to add this term to $F[I+1]$ and subtract it from $F[P]$. Then it is easily seen that both the DP tables $\mathsf{A}_{\text{pref}}[I,J]$ and $\mathsf{A}_{\text{suf}}[I,J]$ and the table $F[K = 1..N]$ can be computed exactly in $O(m^2N^2)$ time and $O(mN)$ space, assuming that f_m and f_a are $O(1)$ time functions and f_{ul} and f_{ur} are $O(N)$ time functions. □

A few simple remarks are in order: The probability $f(D|H \setminus \{H_K\})$ we wish to compute can be extracted from table F by adding up the region $F[1..K]$, which can be done for all K in $O(N)$ time.

If we only wish to compute an ϵ-approximation $\widetilde{f}$, for some consecutive range of m different K values, one can compute these m probabilities $f(D|H \setminus \{H_K\})$ for each kind of modification in $O(min(m,N))$ time:

1. To delete a restriction site from H: Total cost = $O(min(m\Delta^3, N))$ for total of m restriction sites.
2. To change the size of a restriction fragment in H by δ: Total cost = $O(min(m\Delta^3, N))$ for changing each of $(m+1)$ restriction fragments.
3. To add T new restriction sites in H: Total cost = $O(min(m\Delta^3 + T\Delta^4, N))$. Note that typically $T \le m$.
4. To compute the first two derivatives

   $$\frac{\partial^d f(D|H)}{\partial F_k^d}, \qquad d = 1,2, \quad \text{and } k = 0,\ldots,m$$

   of $f(D|H)$ relative to each of $m+1$ fragment sizes: Total cost = $O(min(m\Delta^3, N))$.

3.3 Search Algorithm for Haplotypes

The recurrence equations of the previous subsections and the dynamic programming algorithms based on those allow us to efficiently compute the posterior probability for a single possible pair of maps H_1 and H_2 and their modifications

$$\begin{bmatrix} H_1^{(0)} \\ H_2^{(0)} \end{bmatrix} \Rightarrow \begin{bmatrix} H_1^{(1)} \\ H_2^{(1)} \end{bmatrix} \Rightarrow \begin{bmatrix} H_1^{(2)} \\ H_2^{(2)} \end{bmatrix} \Rightarrow \cdots$$

The computationally expensive part of computing the haplotype map algorithm is the search over possible maps H_1 and H_2 in order to find the one with the highest posterior probability.

Initially, we assume that a single genotype map hypothesis H has been computed and it has been determined that H best matches all data. The algorithms to compute such maps have been developed [4, 3] and have been in use for more than five years. The speed of the main algorithm, GenTig, has been improved through an important heuristic stage that relies on geometric hashing to quickly identify the maps that overlap, and can also be used in the context of haplotyping. The time complexity of this geometric-hashing-stage is super-linear and is given as

$$T_H = O(N + M_D^{4/3}), \quad \text{where } M_D = \sum_{j=1}^{M} m_j + 1,$$

i.e., M_D is the total number of fragments in the optical mapping data. We will see that the actual time for this stage T_H is dominated by the remaining computation involving search over possible haplotype pairs H_1 and H_2, unless the genome we are dealing with is much larger than the human genome; see next subsection.

If our initial hypothesis is H, then $H_1^{(0)} = H_2^{(0)} = H$, and at each stage $H_1^{(i)}$ and $H_2^{(i)}$ must then be refined by trying to add or delete restriction sites and by adjusting the distance between restriction sites by doing a gradient optimization of the probability density of all maps for each fragment size. The result is $H_1^{(i+1)}$ and $H_2^{(i+1)}$.

Note that at each hypothesis-recomputation step, trying each new restriction site polymorphism involves modifying H_1 or H_2 by adding or deleting a restriction site from H_1 (or H_2) only, while trying an RFLP involves modifying the same interval in both H_1 and H_2 by adding some δh to H_1 and subtracting the same δh from H_2. In each case both possible "phases" of each polymorphism is to be accounted for, reversing the use of H_1 and H_2 above. Since both phases must be tested and the better scoring one selected, except when adding the first polymorphism to H_1 and H_2, the search process can easily turn in to $2^{O(N)}$.

Note also that if the data cannot allow the phasing to be determined because there are no (or insufficient) data molecules spanning both polymorphisms, both phases (orientations) will score almost the same. This fact is also recorded since it marks a break in the phasing of polymorphisms.

Further note that RFLP polymorphisms are more expensive to score, since in addition to the phasing (whether H_1 or H_2 has the bigger fragment) it is necessary to determine the amount of the fragment size difference for H_1 and H_2 (the δh value), which can be searched for in $O(1)$ expected time, and the constant is essentially logarithmic in the ratio of the expected fragment length to the resolution of optical mapping. More precisely, this step involves trying a number of different multiples of δh values that is logarithmic in the number of total possible values using the well known unimodal function maximization algorithm based on the golden mean ratio. As an example, the total number of δh values required for any fragment can be bounded by about 20 if the resolution of δh is set at 0.1Kb and the largest restriction fragment length is 50Kb; usually, this number is extremely small: just 1 or 2 small δh values are sufficient to verify that no polymorphism exists.

A purely greedy addition of polymorphisms to H_1 and H_2 is not sufficient to get the phases correct as the search can get stuck in local maxima when two or more polymorphisms are nearby. We avoid this problem by using a heuristic look ahead distance of w restriction sites, and scoring all combinations of polymorphisms in this window, before committing the best scoring set of polymorphisms in H_1 and H_2. With a sufficiently large window size w, the fraction of the polymorphic sites the algorithm misses or phases incorrectly can be made negligible. Since this heuristic can increase the worst case complexity of the algorithm exponentially with the window size w we heuristically determine the smallest possible window w by using simulated data and search the space of possible polymorphisms within a window by adding/deleting just one or two polymorphisms at a time until no further improvement in the probability density occurs.

The overall algorithm must try every possible restriction site and fragment as a possible polymorphic SNP or RFLP respectively using a rolling window of size w restriction sites. This process must be repeated a few times until no further polymorphisms are detected. Typically just two to three iterations of scanning all restriction sites suffice.

3.4 Complexity and Algorithm improvements

The overall complexity of the basic haplotype search algorithms described here, just using the basic DP algorithm from Theorem 3.1, is

$$\begin{aligned}
&\text{Time Complexity}\\
&= \sum_{j=1}^{M} \text{Time to compute } f(D_j|H)\\
&\quad + \sum_{i=1}^{3N}\sum_{j=1}^{M} \text{Time to compute } f(D_j|H_1^{(i)})\\
&\qquad \text{and } f(D_j|H_2^{(i)})\\
&= O\left(\sum_{j=1}^{M} m_j + N\right) + 3N \times O\left(\sum_{j=1}^{M} m_j + N\right)\\
&= O(M_D^2/C + N),
\end{aligned}$$

where C = coverage and $C = (1/N)\sum_{j=1}^{M} m_j$

A couple of simple tricks can be used to significantly speed up the evaluation of conditional probabilities. First H_1 and H_2 are typically only being modified in a single location at a time. If a data map D_j did not previously overlap H_1 or H_2 anywhere near the location we are modifying, we can simply reuse its previous conditional probability density values $f(D_j|H_1)$ and $f(D_j|H_2)$. Since the average number of DNA fragments overlapping any point of the genome is C, a number considerably less than the total number of fragments

M, this makes the total cost of the search for the best H_1 and H_2 asymptotically $O(M_D m)$.

The algorithm can be improved further if we use the dual DP tables from Theorem 3.2, in which m consecutive changes to H_1 and H_2 can be tested in just three times the time it previously took to test just one change to H_1 and H_2. In this case we will recompute the dual DP algorithm for the approximately $2C$ DNA fragments at a time, where these are just the fragments that overlap m consecutive sites. Hence the speedup from switching to the dual DP tables is approximately $m/6$ resulting in an asymptotic complexity of $O(M_D)$

4 Empirical Results

4.1 Haplotype mapping of 22 chromosomes of *T. Pseudoana*

Optical Mapping data was previously collected by Dr. Shiguo Zhou of University of Wisconsin in order to assemble a normal NheI restriction map of the microorganism *T. Pseudoana* (Diatom). To test the haplotyping algorithms described above, we selected the 22 largest chromosomes (out of 25) and separated out the data for each chromosome in order to be able to run the haplotyping algorithm on 22 separate machines. For all except chromosome 19, the algorithm was able to successfully phase all polymorphisms and generate two separate maps. Chromosome 19 has all of its polymorphisms near the two ends of the chromosome and there were not enough molecules that spanned the chromosome end to end to allow their relative phase to be determined.

4.2 Haplotype mapping of 120Mbase region of human genome

We also selected a subset of the Human Optical Mapping data collected by Mr. Alex Lim of University of Wisconsin to assemble the first genome wide SwaI restriction map of the Human genome. The current data set provides an average of 12× data redundancy over the entire human genome, which is insufficient to reliably assemble a genome wide map. Moreover the typical molecule size of 500Kb was shorter than assumed by our simulation. However we selected data that had assembled into a 120Mbase contig and was identified as part of chromosome 4 on the basis of alignment with sequence published by NIH. Even though 12× data redundancy is only sufficient to assemble a haplotype map with low reliably (see next section) chromosome 4 is known to have 60% more SwaI restriction sites than the rest of the human genome which, increases the likelihood that the typical 500Kb molecule in the data can span two to three restriction site polymorphisms. Therefore we attempted to reassemble the data using the haplotype assembly algorithm. The program found 233 restriction site polymorphisms and 12 fragment length polymorphisms, and was able to phase all polymorphisms into 2 contiguous regions. Using simulation studies (see next section), we have determined that one needs to wait until about 50× data redundancy is available when a reliable map can be constructed for this region as well as the rest of the human genome.

4.3 Simulated Data

With real data it is not easy to determine what fraction of false positive or false negative polymorphisms is present in the final map since the true haplotype map is not known independently of our method. To check if this is a problem, we generated simulated data approximating the first 5 mega bases of human chromosome 21, and the typical error rates for Optical Mapping data based on maximum likelihood estimates from previous microbial maps [15, 18, 25] (The more recent data used in the previous sections had maximum likelihood error estimates about 30% better, so the estimates used here are conservative). The simulated data was assembled using different amounts of simulated data corresponding to data redundancy of 6×, 12×, 16×, 24×, 50× and 100× (per haplotype). The results are summarized in Table 1. To understand these numbers consider row 4 corresponding to 16× redundancy). The last column shows that we used 80 molecules in the simulation. Of these 80 molecules the software classified 71 molecules into one of the two haplotype variants. In fact the software made 2 errors and correctly classified only 69 molecules. By comparing the two consensus maps generated by the software we created a list of restriction sites classified as polymorphic (i.e. a SNP was claimed by the software to exist at a restriction site) and this list was then compared with the correct list of SNPs generated from the true in-silico maps. The column with the header "fp SNPs" shows the number of false-positive SNPs claimed by the software. The column with the header "fn SNPs" shows the corresponding number of false-negative SNPs. Similarly for RFLPs (i.e. fragment size polymorphisms due to the simulated insertions/deletions of 3Kb).

5 Discussions and Future Work

Single molecule mapping technologies, such as Optical Mapping, are ideal for detecting genetic markers with phasing information and without population-based assumptions. We

Redundancy	fp SNPs	fn SNPs	fp RFLPs	fn RFLPs	Phase err	Molecules
6x	5	5	1	18	7/26	30
12x	4	2	4	16	2/55	60
16x	2	1	0	12	2/71	80
24x	2	1	1	11	3/111	120
50x	0	1	1	5	4/228	250
100x	0	0	2	1	2/441	500

Figure 2: Haplotyping algorithm performance for 16 SNPs and 24 RFLPs.

formulated an abstraction of the haplotype map assembly problem for all such single molecule mapping technologies and provide a probabilistic linear time algorithm to assemble haplotype maps by combining single molecule restriction maps of long genomic DNAs of average length at least 500Kb containing 2 or more heterozygous polymorphic restriction sites on average.

Single molecule mapping technologies have many advantages over SNP based approaches; we enumerate four. First, restriction maps can reveal not only SNPs that coincide with the restriction sites, but also micro-insertions and deletions, global rearrangements or hemizygous deletions. Even though there is only about 1 micro-insertion or deletion for every 12 SNPs, the average size of a micro-insertion or deletion is over 36 base pairs, and hence accounts for over 75% of all base pair differences vs. just 25% for SNPs [6]. Second, since single molecule methods work directly with genomic DNA and do not require the use of PCR, with such single-molecule methods one can identify markers in repeat regions, segmental duplications, SINES, LINES etc.— regions occupying almost half of the human genome. Third, since single DNA molecule segments are mapped using fluorescent microscopy, this approach is capable of very high throughput (limited primarily by the digital camera throughput) requiring very little DNA, and costs a fraction of the comparable cost for the least expensive SNP based approaches. Of course such a individual haplotype map will reveal only those polymorphic markers (including SNPs) that coincide with restriction sites, but this can be overcome by collecting maps for multiple restriction enzymes: Based on the known SNPs and extrapolating to an estimated minimum 10 million SNPs, then using 4 restriction enzymes with average fragment sizes of 2–4Kb it is possible to detect approximately 200,000 SNPs plus an estimated 130,000 other polymorphisms. This can be extended to about 1 million SNPs and 650,000 other polymorphisms by using about 50 methylation insensitive restriction enzymes in 20 groups. Finally, by suitable choice of restriction enzymes, Optical Mapping is also capable of detecting the epigenetic state in the form of methylated CpG bases which are resistant to many restriction enzymes. Epigenetic information is known to play a key role in genetic diseases like cancer and explains why identical twins may display different genetic traits even though they share the same genetic code.

We estimate that our approach is currently the only approach that can produce a genome wide individual haplotype map for under $1000 (based on 8 restriction enzyme haplotype maps). The dominant SNP based approach requires testing of about 300,000 SNPs which costs at least ten times more per person.

Our approach can be applied to other single molecule mapping technologies. When applied to single molecule technologies to map short 6–8bp LNA hybridization probes, it can be used to sequence the entire human genome: With 50× coverage the location of probes can be determined to within about 200bp. Hence well known error tolerant SBH (Sequencing by Hybridization) algorithms [11] can be used to determine the sequence within any 200bp window from maps of a universal set of about 2048 probes of 6bp, allowing a draft quality individual haplotype sequence to be assembled for about $20,000.

Software

HapTig software will be available soon at `http://www.bioinformatics.cims.nyu.edu/~mishra`

Acknowledgements

We would like to thank Alex Lim and Shiguo Zhou for collecting the human and microbial Optical Mapping data respectively and thank David Schwartz for giving us the opportunity to test our software on this data. We also thank Iuliana Ionita, Vineet Bafna, and Nick Patterson for discussions of similar topics that stimulated our research. The work reported in this paper was supported by grants from NSF's Qubic program, NSF's ITR program, Defense Advanced Research Projects Agency (DARPA), Howard Hughes Medical Institute (HHMI) biomedical support

research grant, the US Department of Energy (DOE), the US air force (AFRL), National Institutes of Health (NIH) and New York State Office of Science, Technology & Academic Research (NYSTAR).

References

[1] D. Altshuler *et al.*, "The Structure of Haplotype Blocks in the Human Genome," *Science*, **296**(5576):2225–9, 2002.

[2] T. Anantharaman *et al.*, "Genomics via Optical Mapping II: Ordered Restriction Maps," *Journal of Computational Biology*, **4**(2):91–118, 1997.

[3] T. Anantharaman *et al.*, "Genomics via Optical Mapping III: Contiging Genomic DNA and Variations," *Proceedings 7th Intl. Cnf. on Intelligent Systems for Molecular Biology*: ISMB '99 , 7:18–27, AAAI Press, 1999

[4] T. Anantharaman *et al.*, "A Probabilistic Analysis of False Positives in Optical Map Alignment and Validation," *Algorithms in Bioinformatics*, First International Workshop, WABI 2001 Proceedings, **LNCS 2149**:27–40, Springer-Verlag, 2001

[5] V. Bafna *et al.*, "Haplotyping as Perfect Phylogeny: A direct approach", Technical Report UCDavis CSE-2002-21,July 17, 2002.

[6] R. Britten *et al.*, "Majority of Divergence between Closely Related DNA Samples is due to Indels," *PNAS*, **100**(8):4461–4465, April 2003.

[7] M. Daly *et al.*, "High-Resolution Haplotype Structure in the Human Genome," *Nat. Genet.*, **29**: 229–232, 2001.

[8] D. Goldstein *et al.*, "Population Genomics: Linkage Disequilibrium holds the Key," *Curr. Biol*, **11**: R576–579, 2001.

[9] D. Gusfield, "Haplotyping as Perfect Phylogeny: Conceptual Framework and Efficient Solutions", RECOMB 2002, April 2002. `http://wwwcsif.cs.ucdavis.edu/~gusfield/acmhapre.pdf`

[10] E. Halperin *et al.*, "Large Scale Recovery of Haplotypes from Genotype Data using Imperfect Phylogeny," Technical report no. UCB/CSD-2-1195, August 2002.

[11] E. Halperin *et al.*, "Handling Long Targets and Errors in Sequencing by Hybridization," *Journal of Computational Biology*, **10**:3–4, 2003

[12] S. Istrail*et al.*, "Methods for Inferring Block-Wise Ancestral History from Haploid Sequences," WABI 2002: 44–59.

[13] R. Karp *et al.*, "Algorithms for Optical Mapping," *Proceedings of RECOMB*, 1998.

[14] R. Karp *et al.*, "Efficient Reconstruction of Haplotype Structure via Perfect Phylogeny," Technical report no. UCB/CSD-2-1196, August 2002.

[15] Z. Lai *et al.*, "A Shotgun Sequence-Ready Optical Map of the Whole *Plasmodium falciparum* Genome," *Nature Genetics*, **23**(3): 309–313, 1999.

[16] G. Lancia *et al.*, "Practical Algorithms and Fixed-Parameeter Tractability for the Single Individual SNP Haplotyping Problem," *WABI 2002*: 29–43.

[17] E. Lander *et al.*, "Linkage Disequilibrium in the Human Genome," *Nature*, **411**: 199–204, 2001.

[18] A. Lim *et al.*, "Shotgun Optical Maps of the Whole *Escherichia coli* O157:H7 Genome", *Genome Research*, **11**(9): 1584–93, 2001.

[19] T. Liu *et al.*, "Bayesian Haplotype Inference for Multiple Linked Single-Nucleotide Polymorphisms," *Am. J. Hum. Genet.*, **70**:157–169, 2002.

[20] B. Mishra and L. Parida, "Partitioning Single-Molecule Maps into Multiple Populations: Algorithms And Probabilistic Analysis," *Discrete Applied Mathematics* (The Computational Molecular Biology Series), **104**(l-3):203-227, August, 2000.

[21] M. Stephens *et al.*, "Statistical Method for Haplotype Reconstruction from Population Data," *American Journal of Human Genetics*, **68**: 978–989, 2001

[22] M. Waterman and V. Dancik, "Simple Maximum Likelihood Methods for the Optical Mapping Problem", *Proceedings of the Workshop of Genome Informatics*, 1997.

[23] M. Waterman *et al.*, "Estimation for Restriction Sites Observed by Optical Mapping Using Markov Chain Monte Carlo", *Journal of Computational Biology*, **5**(3):505–15, 1998.

[24] M. Waterman *et al.*, "Haplotype Reconstruction from SNP Alignment," *RECOMB2003*: 207–216, 2003

[25] S. Zhou *et al.*, "A Whole-Genome Shotgun Optical Map of Yersinia pestis Strain KIM," *Appl. Environ. Microbiol.* **68**(12):6321–6331, 2002.

APPENDIX

The goal of the various DP programming formulations in this paper have been to compute $f(D|H)$, or $f(D|H')$ when H is perturbed to yield H', thus yielding the conditional probabilities of a data map D given a hypothesized consensus map H. This conditional term can be written as a summation over all possible (mutually exclusive) alignments between the particular D and H, and for each alignment the probability density is based on a straightforward enumeration of the map errors implied by the alignment. The key to reasonably fast evaluation of the probability densities summed over all alignments is the use of a dynamic programming recurrence equation, which is equivalent to factoring out the common

sub-expressions of the probability densities across the different alignments. First consider a single arbitrary alignment between a particular D and H. The data map D can be described by a vector of locations of restriction sites

$$D = \left\langle D_J \right\rangle_{J=0}^{m+1},$$

where for convenience the first entry $D[0]$ is always 0 and the last entry $D[m+1]$ is the total size of the map. For notational convenience we may also refer to the entries of this array as $D[J]$ or its subarrays as $D[J..Q]$, where $0 \le J \le Q \le m+1$. Similarly the hypothesis map H will be described by a vector of restriction sites

$$H = \left\langle H_I \right\rangle_{I=0}^{N+1},$$

with analogous representations of $H[I]$ and $H[I..P]$ with $0 \le I \le P \le N+1$. An arbitrary alignment can be described as a list of pairs of restriction sites from H and D that describe which restriction site from H is aligned with which restriction site from D. An example appears in Figure 1.

The expression for the conditional probability density of any alignment as defined in Figure 1. can be written as the product of a term corresponding to the region of alignment between each pair of aligned sites, plus one term for the unaligned region at each end of the alignment.

Let

$$\begin{aligned} P_d &= \text{the digest rate,} \\ \lambda &= \text{the false-positive site rate,} \\ \sigma^2 h &= \text{the Gaussian sizing error variance for a fragment of size } h, \\ P_\nu &= \text{the probability of missing a fragment of unit size, and} \\ R_e &= \text{the breakage rate of DNA (the inverse of the expected fragment size).} \end{aligned}$$

For a random variable x following a Gaussian distribution $\mathcal{N}(\mu, \sigma^2)$, the probability density value at d is

$$G_{\mu,\sigma^2}(d) = \frac{\exp[-(d-\mu)^2/2\sigma^2]}{\sqrt{2\pi}\sigma}.$$

For an aligned region that is not a missing fragment (e.g., alignments (I, J) and (P, Q), such that $P > I$ and $Q > J$) this probability density will be denoted by a function of the form $f_a(I, J, P, Q)$, which will depend on the specific errors in the corresponding region of the alignment between D and H.

$$\begin{aligned} f_a(I,J,P,Q) &= \lambda^{Q-J-1} P_d (1-P_d)^{P-I-1} (1-P_\nu)^{H_P - H_I} \\ &\quad \times G_{(H_P-H_I),\sigma^2(H_P-H_I)}(D_Q - D_J). \end{aligned}$$

Similarly for an aligned region that corresponds to a consecutive number of missing fragments the probability density will be denoted by a function $f_m(I, P)$ (e.g., (I, J) and $(I+1, J)$ will correspond to $f_m(I, I+1)$).

$$f_m(I,P) = P_\nu^{H_P - H_I}.$$

Finally for the probability density of the unaligned portion on the left and right end of each alignment, we shall use $f_{ur}(I, J)$ on the right end if (I, J) is the rightmost aligned pair, and $f_{ul}(I, J)$ on the left end if (I, J) is the leftmost aligned pair. These in turn are computed in terms of the auxiliary functions f_r and f_l:

$$\begin{aligned} &f_r(I,J,P,Q) \\ &= \lambda^{m-J}(1-P_d)^{P-I-1}(1-P_\nu^{H_P-H_I}) \\ &\quad \Big[R_e \Psi(D_{m+1} - D_J, H_P - H_I, H_P - H_Q) \\ &\quad + \mathbb{I}_{P=N+1} G_{(H_{N+1}-H_I),\sigma^2(H_{N+1}-H_I)}(D_{m+1} - D_J) \Big]. \end{aligned}$$

and

$$\begin{aligned} &f_l(I,J,P,Q) \\ &= \lambda^{J-1}(1-P_d)^{I-P-1}(1-P_\nu^{H_I-H_P}) \\ &\quad \Big[R_e \Psi(D_J, H_I - H_P, H_Q - H_P) \\ &\quad + \mathbb{I}_{P=0} G_{(H_I),\sigma^2(H_I)}(D_J) \Big]. \end{aligned}$$

The function Ψ is defined as follows[2]:

$$\begin{aligned} &\Psi(d,h,b) \\ &= \frac{1}{2}\Bigg[\operatorname{erf}\left(\frac{d-h+b}{\sqrt{2\sigma^2[(h-b)\wedge(d\vee h)]}} \right) \\ &\quad + \operatorname{erf}\left(\frac{h-d}{\sqrt{2\sigma^2[(h-b)\wedge(d\vee h)]}} \right) \Bigg] \end{aligned}$$

[2]Notations: $a \wedge b = \max(a, b)$ and $a \vee b = \min(a, b)$.

Now the functions f_{ur} and f_{ul} are defined as follows:

$$f_{ur}(I,J) = \begin{cases} \sum_{P=I+1}^{N+1} f_r(I,J,P,P-1), & \text{if } J \le m; \\ P_\nu^{H_{N+1}-H_N} + R_e \dfrac{P_\nu^{H_{N+1}-H_N}-1}{\log P_\nu}, & \text{if } I = N \ \&\ J = m+1; \\ 0, & \text{otherwise;} \end{cases}$$

and

$$f_{ul}(I,J) = \begin{cases} \sum_{P=0}^{I-1} f_l(I,J,P,P+1), & \text{if } J \ge 1; \\ P_\nu^{H_1} + R_e \dfrac{P_\nu^{H_1}-1}{\log P_\nu}, & \text{if } I = 1 \ \&\ J = 0; \\ 0, & \text{otherwise.} \end{cases}$$

Now all that remains is to write down the probability density of a particular alignment as simply the product of each of the terms f_a, f_m, f_{ul} and f_{ur} while computing A_{suf} recursively as explained in the body. Note that the probability density of any alignment can be broken apart into the product of those terms on either side of any particular alignment pair (I,J). This observation forms the basis of a two-dimensional recurrence using the array $\mathsf{A}_{\text{suf}}[I,J]$, where $1 \le I \le N$ and $0 \le J \le m+1$.

$$\begin{aligned} &\mathsf{A}_{\text{suf}}[I,J] \\ &\quad = f_{ur}(I,J) \\ &\qquad + \mathbb{I}_{I<N} f_m(I,I+1)\mathsf{A}_{\text{suf}}[I+1,J] \\ &\qquad + \sum_{P=I+1}^{N} \sum_{Q=J+1}^{m+1} f_a(I,J,P,Q)\mathsf{A}_{\text{suf}}[P,Q] \\ &f(D|H) \\ &\quad = \sum_{I=1}^{N} \sum_{J=0}^{m+1} f_{ul}(I,J)\mathsf{A}_{\text{suf}}[I,J]. \end{aligned}$$

Next we write down the recurrence equations as we modify H to H'.

We start with an explanation for how to efficiently recompute $f(D|H')$ while deleting one restriction site H_K from H at a time for all possible K, $1 \le K \le N$. The key step is to compute an additional recurrence array A_{pref} which represents the sum of the probability densities of all those alignments between the part of H to the left of site I and the part of D to the left of site J, for which (I,J) is the rightmost aligned pair. The corresponding recurrence equation are shown below:

$$\begin{aligned} &\mathsf{A}_{\text{pref}}[I,J] \\ &\quad = f_{ul}(I,J) \\ &\qquad + \mathbb{I}_{I>0}\mathsf{A}_{\text{pref}}[I-1,J] f_m(I-1,I) \\ &\qquad + \sum_{P=1}^{I-1} \sum_{Q=0}^{J-1} \mathsf{A}_{\text{pref}}[P,Q] f_a(I,J,P,Q) \\ &f_k(D|H) \\ &\quad = \sum_{I=1}^{K-1} \sum_{J=0}^{m+1} \mathsf{A}_{\text{pref}}[I,J] f_{ur}[I,J] + \sum_{I=K+1}^{N} \sum_{J=0}^{m+1} f_{ul}[I,J]\mathsf{A}_{\text{suf}}[I,J] \\ &\qquad + \sum_{J=0}^{m+1} \\ &\qquad \Big[\mathbb{I}_{K<N}\mathsf{A}_{\text{pref}}[K-1,J] f_m(K-1,K+1)\mathsf{A}_{\text{suf}}[K+1,J] \\ &\qquad\quad + \sum_{I=1}^{K-1} \sum_{P=K+1}^{N} \sum_{Q=J+1}^{m+1} \mathsf{A}_{\text{pref}}[I,J] f_a(I,J,P,Q)\mathsf{A}_{\text{suf}}[P,Q] \Big]. \end{aligned}$$

Note that in equation above, none of the terms $\mathsf{A}_{\text{pref}}[I,J]$ or $\mathsf{A}_{\text{suf}}[I,J]$ will change if we remove the restriction site H_K from H. However the terms $f_a(I,J,P,Q)$ will change to $f_a(I,J,P,Q)/(1-P_d)$, and $f_{ur}[I,J]$ and $f_{ul}[I,J]$ will change in a way we will describe below, when H_K is deleted. First we rewrite the previous equation as

$$\begin{aligned} &f_k(D|H) \\ &\quad = \sum_{I=1}^{K-1} \sum_{J=0}^{m+1} \mathsf{A}_{\text{pref}}[I,J] \sum_{P=I+1}^{N+1} f_r(I,J,P,P-1) \\ &\qquad + \sum_{I=K+1}^{N} \sum_{J=0}^{m+1} \mathsf{A}_{\text{suf}}[I,J] \sum_{P=0}^{I-1} f_l(I,J,P,P+1) \\ &\qquad + \sum_{J=0}^{m+1} \\ &\qquad \Big[\mathbb{I}_{K<N}\mathsf{A}_{\text{pref}}[K-1,J] f_m(K-1,K+1)\mathsf{A}_{\text{suf}}[K+1,J] \\ &\qquad\quad + \sum_{I=1}^{K-1} \sum_{P=K+1}^{N} \sum_{Q=J+1}^{m+1} \mathsf{A}_{\text{pref}}[I,J] f_a(I,J,P,Q)\mathsf{A}_{\text{suf}}[P,Q] \Big]. \end{aligned}$$

We now write down the recurrence equation to reflect the deletion of H_K from H and corresponding changes in f_a, f_l

and f_r to get the result:

$$
\begin{aligned}
& f(D|H \setminus \{H_K\}) \\
= & \sum_{I=1}^{K-1}\sum_{J=0}^{m+1} \mathsf{A}_{\mathrm{pref}}[I,J] \sum_{P=I+1}^{N+1} f_r^{-(H_k)}(I,J,P,P-1) \\
& + \sum_{I=K+1}^{N}\sum_{J=0}^{m+1} \mathsf{A}_{\mathrm{suf}}[I,J] \sum_{P=0}^{I-1} f_l^{-(H_k)}(I,J,P,P+1) \\
& + \sum_{J=0}^{m+1} \\
& \Bigg[\mathbb{I}_{K<N}\mathsf{A}_{\mathrm{pref}}[K-1,J] f_m(K-1,K+1)\mathsf{A}_{\mathrm{suf}}[K+1,J] \\
& \quad + \sum_{I=1}^{K-1}\sum_{P=K+1}^{N}\sum_{Q=J+1}^{m+1} \mathsf{A}_{\mathrm{pref}}[I,J]\frac{f_a(I,J,P,Q)}{1-P_d}\mathsf{A}_{\mathrm{suf}}[P,Q]\Bigg],
\end{aligned}
$$

where

$$
\begin{aligned}
& f_r^{-(H_K)}(K,I,J,P) \\
= & \begin{cases} \dfrac{f_r(I,J,P,P-1)}{(1-P_d)}, & \text{if } K < P-1; \\ f_r(I,J,P,P-2), & \text{if } K = P-1; \\ 0, & \text{if } K = P; \\ f_r(I,J,P,P-1), & \text{if } K > P, \end{cases}
\end{aligned}
$$

and analogously

$$
\begin{aligned}
& f_l^{-(H_K)}(K,I,J,P) \\
= & \begin{cases} f_l(I,J,P,P+1), & \text{if } K < P; \\ 0, & \text{if } K = P; \\ f_l(I,J,P,P+2), & \text{if } K = P+1; \\ \dfrac{f_l(I,J,P,P+1)}{(1-P_d)}, & \text{if } K > P+1. \end{cases}
\end{aligned}
$$

Recurrence equation for adding a restriction site H_T to H somewhere between H_{K-1} and H_K is shown below.

$$
\begin{aligned}
& f(D|H \cup \{H_T\}, H_{K-1} < H_T < H_K) \\
= & \sum_{I=1}^{K-1}\sum_{J=0}^{m} \mathsf{A}_{\mathrm{pref}}[I,J] \sum_{P=I+1}^{N+1} f_r^{+(H_T)}(K,I,J,P) \\
& + \sum_{I=K}^{N}\sum_{J=1}^{m+1} \mathsf{A}_{\mathrm{suf}}[I,J] \sum_{P=0}^{I-1} f_l^{+(H_T)}(K,I,J,P) \\
& + \sum_{J=0}^{m+1}\Bigg[\mathsf{A}_{\mathrm{pref}}[K-1,J] f_m(K-1,K+1)\mathsf{A}_{\mathrm{suf}}[K+1,J] \\
& \quad + \sum_{I=1}^{K-1}\sum_{P=K}^{N}\sum_{Q=J+1}^{m+1} \\
& \qquad \mathsf{A}_{\mathrm{pref}}[I,J] f_a(I,J,P,Q)(1-P_d)\mathsf{A}_{\mathrm{suf}}[P,Q]\Bigg] \\
& + \sum_{J=0}^{m+1} \mathsf{A}_{\mathrm{pref}}^{(H_K\to H_T)}[K,J]\mathsf{A}_{\mathrm{suf}}^{(H_K\to H_T)}[K-1,J],
\end{aligned}
$$

where the the notations $\mathsf{A}_{\mathrm{pref}}^{(H_K\to H_T)}[K,J]$ (respectively, $\mathsf{A}_{\mathrm{suf}}^{(H_K\to H_T)}[K-1,J]$) means to evaluate $\mathsf{A}_{\mathrm{pref}}[K,J]$ (respectively, $\mathsf{A}_{\mathrm{suf}}[K-1,J]$) using its defining equation provided previously while replacing any occurrence of H_K with H_T. Note that the equation shown above depends on the exact value of H_T only in the last summation term. Furthermore

$$
\begin{aligned}
& f_r^{+(H_K)}(K,I,J,P) \\
= & \begin{cases} f_r(I,J,P,P-1)(1-P_d), & \text{if } K \le P-1; \\ f_r^{(H_K\to H_T)}(I,J,K,K-1) & \\ \quad + f_r^{(H_{K-1}\to H_T)}(I,J,K,K-1), & \text{if } K = P; \\ f_r(I,J,P,P-1), & \text{if } K > P, \end{cases}
\end{aligned}
$$

and analogously

$$
\begin{aligned}
& f_l^{+(H_K)}(K,I,J,P) \\
= & \begin{cases} f_l(I,J,P,P+1), & \text{if } K \le P. \\ f_l^{(H_K\to H_T)}(I,J,K-1,K) & \\ \quad + f_l^{(H_{K-1}\to H_T)}(I,J,K-1,K), & \text{if } K = P+1; \\ f_l(I,J,P,P+1)(1-P_d), & \text{if } K \le P-1. \end{cases}
\end{aligned}
$$

Recurrence equation for adding a small amount δh to one restriction fragment $[H_K, H_{K+1}]$. Note that this is equivalent to changing

$$
\begin{aligned}
& \langle H_1,\ldots,H_{K-1},H_K,H_{K+1},\ldots,H_N\rangle \\
& \quad \to \quad \langle H_1,\ldots,H_{K-1},H_K,H_{K+1}+\delta h,\ldots,H_N+\delta h\rangle.
\end{aligned}
$$

$$
\begin{aligned}
& f(D|H : \forall_{T>K} H_T \to H_T + \delta h) \\
= & \sum_{I=1}^{K}\sum_{J=0}^{m} \mathsf{A}_{\mathrm{pref}}[I,J] \sum_{P=I+1}^{N+1} f_r^{(+\delta h)}(K,I,J,P) \\
& + \sum_{I=K+1}^{N}\sum_{J=1}^{m+1} \mathsf{A}_{\mathrm{suf}}[I,J] \sum_{P=0}^{I-1} f_l^{(+\delta h)}(K,I,J,P) \\
& + \sum_{J=0}^{m+1} \\
& \Bigg[\mathbb{I}_{K\le N}\mathsf{A}_{\mathrm{pref}}[K,J] P_\nu^{\delta h} f_m(K,K+1)\mathsf{A}_{\mathrm{suf}}[K+1,J] \\
& \quad + \sum_{I=1}^{K-1}\sum_{P=K}^{N}\sum_{Q=J+1}^{m+1} \mathsf{A}_{\mathrm{pref}}[I,J] f_a^{(+\delta h)}(I,J,P,Q)\mathsf{A}_{\mathrm{suf}}[P,Q]\Bigg],
\end{aligned}
$$

where

$$
\begin{aligned}
& f_a^{(+\delta h)}(I,J,P,Q) \\
&\quad = \quad f_a^{(H_P \to H_P+\delta h)}(I,J,P,Q) \\
& f_r^{(+\delta h)}(K,I,J,P) \\
&\quad = \quad \begin{cases} f_r^{H_{P-1}\to H_{P-1}+\delta h, H_P \to H_P+\delta h}(I,J,P,P-1), \\ \qquad \text{if } K < P-1; \\ f_r^{H_P \to H_P+\delta h}(I,J,P,P-1), \\ \qquad \text{if } K = P-1; \\ f_r(I,J,P,P-1), \\ \qquad \text{if } K \geq P, \end{cases}
\end{aligned}
$$

and analogously

$$
\begin{aligned}
& f_l^{(+\delta h)}(K,I,J,P) \\
&\quad = \quad \begin{cases} f_l(I,J,P,P+1), \\ \qquad \text{if } K < P; \\ f_l^{H_P \to H_P-\delta h}(I,J,P,P+1), \\ \qquad \text{if } K = P; \\ f_l^{H_P \to H_P-\delta h, H_{P+1}\to H_{P+1}-\delta h}(I,J,P,P+1) \\ \qquad \text{if } K > P. \end{cases}
\end{aligned}
$$

Finally the first two ($d = 1,\ 2$) partial derivatives of $f(D|H)$ relative to all fragment sizes $F_K = H_{K+1} - H_K$, $0 \leq K \leq N$, can be computed by using the recurrence equation shown below.

$$
\begin{aligned}
& \frac{\partial^d f(D|H)}{\partial F_K^d} \\
&= \sum_{I=1}^{K}\sum_{J=0}^{m} \mathsf{A}_{\text{pref}}[I,J] \sum_{P=I+1}^{N+1} \frac{\partial^d f_r(I,J,P,P-1)}{\partial F_K^d} \\
&\quad + \sum_{I=K+1}^{N}\sum_{J=1}^{m+1} \mathsf{A}_{\text{suf}}[I,J] \sum_{P=0}^{I-1} \frac{\partial^d f_l(I,J,P,P+1)}{\partial F_K^d}] \\
&\quad + \mathbb{I}_{K=N} \mathsf{A}_{\text{pref}}[N,m+1] \frac{\partial^d f_m^{(N)}}{\partial F_N^d} \\
&\quad + \mathbb{I}_{K=0} \mathsf{A}_{\text{suf}}[1,0] \frac{\partial^d f_m^{(0)}}{\partial F_0^d} \\
&\quad + \sum_{J=0}^{m+1} \\
&\quad \Bigg[\mathbb{I}_{K<N} \mathsf{A}_{\text{pref}}[K,J] \frac{\partial^d f_m(K,K+1)}{\partial F_K^d} \mathsf{A}_{\text{suf}}[K+1,J] \\
&\qquad + \sum_{I=1}^{K}\sum_{P=K+1}^{N}\sum_{Q=J+1}^{m+1} \mathsf{A}_{\text{pref}}[I,J] \frac{\partial^d f_a(I,J,P,Q)}{\partial F_I^d} \mathsf{A}_{\text{suf}}[P,Q] \Bigg].
\end{aligned}
$$

The differential expressions in the recurrence equations are computed as shown in the following formulæ.

$$
\begin{aligned}
\frac{\partial^d f_m(K,K+1)}{\partial F_K^d} &= f_m(K,K+1)(\log P_\nu)^d. \\
\frac{\partial^d f_m^{(N)}}{\partial F_N^d} &= f_m(N,N+1)(R_e + \log P_\nu)(\log P_\nu)^{d-1}. \\
\frac{\partial^d f_m^{(0)}}{\partial F_0^d} &= f_m(0,1)(R_e + \log P_\nu)(\log P_\nu)^{d-1},
\end{aligned}
$$

Next, we derive

$$
\begin{aligned}
& \frac{\partial f_a(I,J,P,Q)}{\partial F_I} = f_(I,J,P,Q) \Bigg[\\
&\quad \left\{ \frac{G'(D_Q - D_J, H_P - H_I)}{G'(D_Q - D_J, H_P - H_I)} - \frac{f_m(I,P)\log P_\nu}{1 - f_m(I,P)} \right\} \Bigg] \\
& \frac{\partial^2 f_a(I,J,P,Q)}{\partial F_I^2} = f_(I,J,P,Q) \Bigg[\\
&\quad \left\{ \frac{G'(D_Q - D_J, H_P - H_I)}{G(D_Q - D_J, H_P - H_I)} - \frac{f_m(I,P)\log P_\nu}{1 - f_m(I,P)} \right\}^2 \\
&\quad + \frac{G''(D_Q - D_J, H_P - H_I)}{G(D_Q - D_J, H_P - H_I)} - \left(\frac{G'(D_Q - D_J, H_P - H_I)}{G(D_Q - D_J, H_P - H_I)} \right)^2 \\
&\quad - \frac{f_m(I,P)\log P_\nu^2}{1 - f_m(I,P)^2} \Bigg].
\end{aligned}
$$

where

$$
\begin{aligned}
G(d,h) &= \frac{\exp[-(d-h)^2/2\sigma^2 h]}{\sqrt{2\pi\sigma^2 h}} \\
G'(d,h) &= \left(\frac{d^2 - h^2 - \sigma^2 h}{2\sigma^2 h^2} \right) G(d,h) \\
G''(d,h) &= \left[\left(\frac{d^2 - h^2 - \sigma^2 h]}{2\sigma^2 h^2} \right)^2 - \frac{d^2}{\sigma^2 h^3} + \frac{1}{2h^2} \right] G(d,h)
\end{aligned}
$$

Furthermore,

$$
\begin{aligned}
& \frac{\partial f_r(I,J,P,P-1)}{\partial F_K} \\
&\quad = \quad \begin{cases} f_r^{(a,1)}(I,J,P) - f_r^{(b,1)}(I,J,P), & \text{if } K < P-1; \\ f_r^{(a,1)}(I,J,P), & \text{if } K = P-1; \\ 0, & \text{if } K \geq P, \end{cases}
\end{aligned}
$$

and analogously

$$
\begin{aligned}
& \frac{\partial f_l(I,J,P,P-1)}{\partial F_K} \\
&\quad = \quad \begin{cases} 0, & \text{if } K < P, \\ f_l^{(a,1)}(I,J,P), & \text{if } K = P; \\ f_l^{(a,1)}(I,J,P) - f_l^{(b,1)}(I,J,P), & \text{if } K > P; \end{cases}
\end{aligned}
$$

Next, we consider the second derivatives:

$$\frac{\partial^2 f_r(I,J,P,P-1)}{\partial F_K^2} = \begin{cases} f_r^{(a,2)}(I,J,P) - f_r^{(b,2)}(I,J,P), & \text{if } K < P-1; \\ f_r^{(a,2)}(I,J,P), & \text{if } K = P-1; \\ 0, & \text{if } K \geq P, \end{cases}$$

and analogously

$$\frac{\partial^2 f_l(I,J,P,P-1)}{\partial F_K^2} = \begin{cases} 0, & \text{if } K < P, \\ f_l^{(a,2)}(I,J,P), & \text{if } K = P; \\ f_l^{(a,2)}(I,J,P) - f_l^{(b,2)}(I,J,P), & \text{if } K > P; \end{cases}$$

The terms $f_r^{(a,1)}$, $f_r^{(a,2)}$, $f_l^{(a,1)}$, and $f_l^{(a,2)}$ are defined as shown:

$$\begin{aligned} & f_r^{(a,1)}(I,J,P) \\ &= \lambda^{m-J}(1-P_d)^{P-I-1}\Big\{ \\ &\quad (1-f_m(I,P))\Big[\\ &\qquad R_e\Psi_a(D_{m+1}-D_J, H_P-H_I, H_{P-1}-H_I) \\ &\qquad +\mathbb{I}_{P>N}G'(D_Q-D_J, H_P-H_I)\Big] \\ &\quad -f_m(I,P)\log P_\nu\Big[\\ &\qquad R_e\Psi(D_{m+1}-D_J, H_P-H_I, H_{P-1}-H_I) \\ &\qquad \mathbb{I}_{P>N}G(D_Q-D_J, H_P-H_I)\Big]\Big\} \end{aligned}$$

and

$$\begin{aligned} & f_r^{(a,2)}(I,J,P) \\ &= \lambda^{m-J}(1-P_d)^{P-I-1}\Big[\\ &\qquad R_e\Psi'_a(D_{m+1}-D_J, H_P-H_I, H_{P-1}-H_I) \\ &\qquad +\mathbb{I}_{P>N}G''(D_Q-D_J, H_P-H_I)\Big] \end{aligned}$$

Analogously,

$$\begin{aligned} & f_l^{(a,1)}(I,J,P) \\ &= \lambda^{J-1}(1-P_d)^{I-P-1}\Big\{ \\ &\quad (1-f_m(P,I))\Big[\\ &\qquad R_e\Psi_a(D_J, H_I-H_P, H_I-H_{P+1}) \\ &\qquad +\mathbb{I}_{P=0}G'(D_J, H_I-H_P)\Big] \\ &\quad -f_m(P,I)\log P_\nu\Big[\\ &\qquad R_e\Psi(D_J, H_I-H_P, H_I-H_{P+1}) \\ &\qquad \mathbb{I}_{P=0}G(D_J, H_I-H_P)\Big]\Big\} \end{aligned}$$

and

$$\begin{aligned} & f_l^{(a,2)}(I,J,P) \\ &= \lambda^{J-1}(1-P_d)^{I-P-1}\Big[\\ &\qquad R_e\Psi'_a(D_J, H_I-H_P, H_I-H_{P+1}) \\ &\qquad +\mathbb{I}_{P=0}G''(D_J, H_I-H_P)\Big] \end{aligned}$$

Next, the terms $f_r^{(b,1)}$, $f_r^{(b,2)}$, $f_l^{(b,1)}$, and $f_l^{(b,2)}$ are defined as shown:

$$\begin{aligned} & f_r^{(b,1)}(I,J,P) \\ &= \lambda^{m-J}(1-P_d)^{P-I-1}(1-f_m(I,P)) \\ &\qquad R_e\Psi_b(D_{m+1}-D_J, H_P-H_I, H_{P-1}-H_I) \end{aligned}$$

and

$$\begin{aligned} & f_r^{(b,2)}(I,J,P) \\ &= \lambda^{m-J}(1-P_d)^{P-I-1} \\ &\qquad R_e\Psi'_b(D_{m+1}-D_J, H_P-H_I, H_{P-1}-H_I) \end{aligned}$$

Analogously,

$$\begin{aligned} & f_l^{(b,1)}(I,J,P) \\ &= \lambda^{J-1}(1-P_d)^{I-P-1}(1-f_m(P,I)) \\ &\qquad R_e\Psi_b(D_J, H_I-H_P, H_I-H_{P+1}) \end{aligned}$$

and

$$\begin{aligned} & f_l^{(b,2)}(I,J,P) \\ &= \lambda^{J-1}(1-P_d)^{I-P-1} \\ &\qquad R_e\Psi'_b(D_J, H_I-H_P, H_I-H_{P+1}) \end{aligned}$$

Here the functions Ψ_a, Ψ'_a, Ψ_b, and Ψ'_b have the following definitions.

$$\begin{aligned}
\Psi_a(d, h_1, h_2) &= \frac{\exp[-(d-h_1)^2/2\sigma^2(d \vee h_1) \wedge h_2]}{\sqrt{2\pi\sigma^2(d \vee h_1) \wedge h_2}} \\
\Psi'_a(d, h_1, h_2) &= \left(\frac{d-h_1}{2\sigma^2(d \vee h_1) \wedge h_2}\right) \Psi_a(d, h_1, h_2) \\
\Psi_b(d, h_1, h_2) &= \frac{\exp[-(d-h_2)^2/2\sigma^2(d \vee h_1) \wedge h_2]}{\sqrt{2\pi\sigma^2(d \vee h_1) \wedge h_2}} \\
\Psi'_b(d, h_1, h_2) &= \left(\frac{d-h_2}{2\sigma^2(d \vee h_1) \wedge h_2}\right) \Psi_a(d, h_1, h_2)
\end{aligned}$$