An EM-type Algorithm for Ordered Restriction Map Alignment

Hirotada Kobayashi hirotada@is.s.u-tokyo.ac.jp
Hiroshi Imai imai@is.s.u-tokyo.ac.jp

Department of Information Science, Faculty of Science, University of Tokyo
7-3-1 Hongo, Bunkyo-ku, Tokyo 113-0033, Japan

Abstract

Constructing restriction maps is one of the important steps towards the determination of DNA sequences. Recently, the single-molecule approaches to constructing restriction maps, such as Optical Mapping by D. Schwartz *et al.*, have developed. In practice, with the single-molecule approach like Optical Mapping, the identification of the restriction sites is complicated by several error factors due to resolving power of biological experiments. The ordered restriction map alignment problem is a problem to estimate the actual restriction sites from many imprecise copies of map from single molecule.

In this paper, we formulate the problem on the basis of the statistical maximum likelihood estimate, and propose a new efficient local search algorithm for this problem, by applying the Expectation-Maximization (EM) algorithm along with the concept of two-clustering. Our algorithm works well for a lot of sets of simulated data, some of which we believe more difficult than the actual cases.

1 Introduction

Optical Mapping [15, 2, 16, 11] is a new single-molecule approach to constructing restriction maps, recently developed by D. Schwartz *et al.* In the Optical Mapping, single copies of the target DNA molecule are fluorescently stained, and stretched to attach to a glass support under a microscope. Then restriction enzymes are activated in the medium, and cleave the molecules at their restriction sites. The molecule fragments remain on the surface, but the elasticity of linearized DNA pulls back the molecule ends at the new cleaved sites. Therefore we can identify the restriction sites as gaps under the microscope in the fluorescent line of the molecule, and the length of each fragment can be measured based on the fluorescent intensity of it.

Thus, in this case the relative order of the fragments is not lost, and, in principle, the restriction map of the whole molecule is obtained. However, the obtained map is imprecise due to several experimental errors. First, not all of restriction sites are digested in each molecule and the experimentally obtained map may have some other hidden restriction sites (*false negative errors*). Second, to the contrary, some of the gaps detected in the experiment may not be the actual restriction sites (*false positive errors*). Third, errors in the measured length of each fragment are not avoidable (*sizing errors*). The effect of these three errors seems to be removed by gathering experimental data from many molecules, however, using many molecules causes other problems. The main difficulty is that the orientation of each molecule (left to right or vice versa) is not known, and the correct orientation of each molecule should be determined in order to obtain the correct map. Besides them, there may be a spurious data among the gathered data, or some of the molecule fragments may be missed under the experiment, etc. Thus we should solve some other mathematical problems to build a map.

The *ordered restriction map alignment problem* is a problem to estimate the actual restriction map from the gathered imprecise maps of the same DNA molecule by a single-molecule approach like Optical Mapping.

Roughly speaking, there are two types of approaches to this problem, statistical approaches [1, 4, 9] and combinatorial approaches [12, 13, 14, 6, 7].

As for the combinatorial approaches, the major drawback lies in the weakness of their statistical background, especially for sizing errors, with respect to the criteria to be optimized. Moreover, their algorithms put an unrealistic assumption that the approximate values of digestion rate or the average number of false cuts per molecule should be known.

On the other hand, as for the statistical approaches, Anantharaman *et al.* [1] are the first that gave a detailed probabilistic model. They compute the Bayesian estimation by a steepest ascent local search. They also defined several simplified models and proved NP-completeness of problems on these simplified models. However, in their local search algorithm, they solve partial differential equations approximately. Thus their local search is not a "steepest" ascent one in a strict sense. Moreover their model is very complicated and is not so easy to handle. Dančík and Waterman [4] formulated the problem using the Gaussian mixture model. Their idea is based on the clustering of the observed restriction sites, and they compute the maximum likelihood estimate by employing the Expectation-Maximization (EM) algorithm for their model. Since their algorithm also puts an unrealistic assumption that the number of the restriction sites should be known before computing, Lee *et al.* [9] extended this idea and applied the Reversible-Jump Markov Chain Monte Carlo to computing maximum likelihood estimate in order to reinforce this weak point. However, their algorithm does not work well when there is a lot of noises on data due to frequent false positives.

In this paper we formulate the problem in a statistical way, incorporating the merit of combinatorial approaches. That is, we use the discretized model as in [12, 7] to simplify our model. However, instead of taking a combinatorial approach itself, we introduce our statistical model, similar to [1], although ours is based on the discretized formulation, and take the maximum likelihood approach.

Generally speaking, in comparison with the Newton-type methods, the EM algorithm is numerically stable with each iteration increasing likelihood, and has reliable global convergence under fairly general conditions, although the algorithm may converge slowly (cf. [10]). Thus we consider applying the EM-type algorithm rather than a steepest ascent method.

We classify the error factors into three types, sizing errors, false negatives or positives, and orientation errors. We remove the noises of first two types of error factors with the EM algorithm, by assuming imaginary state in which only sizing errors take place. Our discretized model enables us to fulfill global maximization in each maximizing step of the EM algorithm in a strict sense. For orientation errors, we utilize the concept of two-clustering.

Since our algorithm above is a local search algorithm and its accuracy greatly depends on the initial solution, we also propose an efficient heuristic algorithm to generate an initial solution. This heuristic algorithm is a greedy-type one, however, it can find a solution of very high quality without knowing any advance informations for the values of the parameters, and by itself can match other algorithms enough.

We also confirmed that our algorithm works well for a lot of sets of simulated data, some of which we believe more difficult than the actual cases.

2 Ordered Restriction Map Alignment Problem

Here we give our formulation of the ordered restriction map alignment problem. Since the location of cut sites in each molecule can be measured only with limited accuracy, and for computational convenience, we take a discretized formulation of the problem. We introduce a probabilistic model on our discretized formulation, and compute the maximum likelihood estimate on this model.

2.1 Discretized Representation of the Molecules

We assume that each observed molecule is described in a 0-1 string of length l. Let n denote the number of the observed molecules, and let $\boldsymbol{s}_1, \cdots, \boldsymbol{s}_n$ denote the observed molecules each represented by a 0-1 string of length l. Let $s_{i,j}$ denote the jth entry of the ith molecule in 0-1 representation ($i = 1, \cdots, n, j = 1, \cdots, l$), where $s_{i,j} = 1$ if and only if there is a cut site in position j of the ith molecule, and otherwise $s_{i,j} = 0$.

In order to deal with the orientation errors, we introduce unobservable 0-1 indicator variables d_i which represent the orientation of the observed molecules ($i = 1, \cdots, n$). If the orientation of the ith molecule is preserved, d_i takes 0, and if it is reversed, d_i takes 1.

For simplicity, we also use a matrix $M(\boldsymbol{d}) = \{M(\boldsymbol{d})_{ij}\}$ where

$$M(\boldsymbol{d})_{ij} = s_{i,j}^{1-d_i} s_{i,l-j+1}^{d_i} \ (i = 1, \cdots, n, j = 1, \cdots, l),$$

where $\boldsymbol{d} = (d_1, \cdots, d_n)^T$. Let $\overline{\boldsymbol{s}}_i$ denote the reverse of $\boldsymbol{s}_i$. The jth entry of $\overline{\boldsymbol{s}}_i$ is defined by $\bar{s}_{i,j} = s_{i,l-j+1}$ ($j = 1, \cdots, l$).

We also use 0-1 representation for the unknown correct restriction map. Let $\boldsymbol{S}$ denote this unknown actual restriction map represented by a 0-1 string of length l. If the actual restriction map $\boldsymbol{S}$ has r restriction sites, and if the locations of r actual restriction sites are $\theta_1, \cdots, \theta_r$ ($\theta_i \in \mathbf{N}, 1 \leq \theta_1 < \cdots < \theta_r \leq l$), the representation of $\boldsymbol{S} = (\theta_1, \cdots, \theta_r)$ precisely provides one 0-1 string of length l with just r 1's. Thus we also use this representation for $\boldsymbol{S}$. Although we gave a discretized representation for $\boldsymbol{S}$ above, in the rest of this paper we actually use the continuous representation of $\boldsymbol{S} = (\theta_1, \cdots, \theta_r)$ where $\theta_i \in \mathbf{R}, 0 < \theta_1 < \cdots < \theta_r \leq l$ for computational convenience, since there is no reason that we should represent the actual map in a discretized way.

As for the error factors, we deal with the following four types of errors: false positives, false negatives, sizing errors, and orientation errors. We assume that these four types of errors occur independently in a probabilistic sense.

Then, roughly speaking, the problem we solve is to find the actual map $\boldsymbol{S}$ and the 0-1 assignment of each d_i that maximize the probability we observe the set of sample data $\boldsymbol{s}_i$ of size n in a given probabilistic model.

2.2 Probabilistic Model for Sizing Errors

Here we give our probabilistic model for sizing errors. For simplicity, in this subsection and after a while, we consider a simplified problem of no orientation errors. We also assume for a while that r, the number of the actual restriction sites, is known. We will see how to estimate r in Section 4.

We assume that the kth actual restriction site at θ_k is observed in each sample data $\boldsymbol{s}_i$ according to the normal distribution with mean θ_k and variance σ_k^2, if it appears as a cut site in $\boldsymbol{s}_i$, and if there occurs no orientation errors ($i = 1, \cdots, n, k = 1, \cdots, r$).

For convenience, let us consider such an imaginary state that only sizing errors have occurred, and false positives or negatives have not occurred yet. Let $\boldsymbol{z}_i$ denote the ith molecule in such an imaginary state represented by a 0-1 string of length l ($i = 1, \cdots, n$). Note that $\boldsymbol{z}_i$ has just r 1's, since there have been no false positives or negatives yet. Let $\tilde{\theta}_{i,k}$ denote the location of the kth restriction site in $\boldsymbol{z}_i$ ($i = 1, \cdots, n, k = 1, \cdots, r$). Then the probability that the kth actual restriction site in $\boldsymbol{S}$ at location θ_k is observed at location $\tilde{\theta}_{i,k}$ in $\boldsymbol{z}_i$ is given by

$$h(\tilde{\theta}_{i,k}; \theta_k, \sigma_k^2) = \begin{cases} \displaystyle\int_{-\infty}^{\tilde{\theta}_{i,k}+\frac{1}{2}} g(t; \theta_k, \sigma_k^2)\, dt & \text{if } \tilde{\theta}_{i,k} = 1, \\ \displaystyle\int_{\tilde{\theta}_{i,k}-\frac{1}{2}}^{\tilde{\theta}_{i,k}+\frac{1}{2}} g(t; \theta_k, \sigma_k^2)\, dt & \text{if } 2 \leq \tilde{\theta}_{i,k} \leq l-1, \\ \displaystyle\int_{\tilde{\theta}_{i,k}-\frac{1}{2}}^{\infty} g(t; \theta_k, \sigma_k^2)\, dt & \text{if } \tilde{\theta}_{i,k} = l, \end{cases}$$

if we take the effect of the discretization of the data into account, where g is a probability density function of normal distribution $\mathrm{N}(\mu, \sigma^2)$ with mean μ and variance σ^2 defined as $g(x; \mu, \sigma^2) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{(x-\mu)^2}{2\sigma^2}}$.

In practice, each $\tilde{\theta}_{i,k}$ is never less than 1 or more than l, and the order of restriction sites should be preserved with sizing errors, that is, $1 \leq \tilde{\theta}_{i,1} < \cdots < \tilde{\theta}_{i,r} \leq l$ should be satisfied for all $i = 1, \cdots, n$. Thus we do not allow the sizing errors to change the order of restriction sites, which was permitted

in the Gaussian mixture model by Dančík and Waterman [4] or Lee *et al.* [9]. Instead, we permit a little abuse of our model in a probabilistic sense, and define the probability that we get $\boldsymbol{z}_i$ from $\boldsymbol{S}$ in the ith molecule with only sizing errors as follows:

$$p_{\text{sizing\_error}}(\boldsymbol{z}_i; \boldsymbol{S}) = \begin{cases} \displaystyle\prod_{k=1}^{r} h(\tilde{\theta}_{i,k}; \theta_k, \sigma_k^2) & \text{if } 1 \le \tilde{\theta}_{i,1} < \cdots < \tilde{\theta}_{i,r} \le l, \\ 0 & \text{otherwise.} \end{cases} \tag{1}$$

By this definition the total sum of $p_{\text{sizing\_error}}(\boldsymbol{z}_i; \boldsymbol{S})$ is less than 1, hence, $p_{\text{sizing\_error}}(\boldsymbol{z}_i; \boldsymbol{S})$ is not a probability in a strict sense. However, as far as we use the model with normal distributions, this contradiction cannot be avoided. In the rest part of this paper, for computational convenience, we approximate h, the probability density function of the discretized version of normal distribution, by g, the probability density function of the original normal distribution.

2.3 Probabilistic Model for False Positives and False Negatives

In this subsection we give a probabilistic model for false positives and negatives.

We assume that false positives only depend on physical factors during the experiment, and that false negatives only depend on partial digestion of the restriction sites. Since we assume the probabilistic independence among four types of errors, we can consider that false positives and negatives occur after sizing errors have taken place.

Let p_{d} denote the *digestion rate*, which is the probability of an actual restriction site to appear as a cut site in each molecule. We assume that the digestion rate is equal for every actual restriction site. Let p_{fp} denote the *false positive rate* per bit, which is the probability of appearing false cut site at each position. Then, in the absence of orientation errors, the probability that we get $\boldsymbol{s}_i$ from $\boldsymbol{z}_i$ with only false positives and negatives is given by

$$p_{\text{fp\&fn}}(\boldsymbol{s}_i; \boldsymbol{z}_i) = \prod_{j=1}^{l} p_{\text{bit\_error}}(s_{i,j}; z_{i,j}, p_{\text{d}}, p_{\text{fp}}),$$

where $z_{i,j}$ denotes the jth entry of $\boldsymbol{z}_i$, and $p_{\text{bit\_error}}(x; y, p_{\text{d}}, p_{\text{fp}})$ is defined as

$$p_{\text{bit\_error}}(x; y, p_{\text{d}}, p_{\text{fp}}) = \begin{cases} 1 - p_{\text{fp}} & \text{if } (x, y) = (0, 0), \\ p_{\text{fp}} & \text{if } (x, y) = (1, 0), \\ (1 - p_{\text{fp}})(1 - p_{\text{d}}) & \text{if } (x, y) = (0, 1), \\ p_{\text{fp}} + (1 - p_{\text{fp}})p_{\text{d}} & \text{if } (x, y) = (1, 1). \end{cases}$$

Note that the following representation for $p_{\text{bit\_error}}(x; y, p_{\text{d}}, p_{\text{fp}})$ is equivalent to above definition:

$$p_{\text{bit\_error}}(x; y, p_{\text{d}}, p_{\text{fp}}) = (1 - p_{\text{fp}})^{(1-x)} p_{\text{fp}}^{x(1-y)} (1 - p_{\text{d}})^{(1-x)y} (p_{\text{fp}} + p_{\text{d}} - p_{\text{fp}} p_{\text{d}})^{xy}.$$

With this representation, $p_{\text{fp\&fn}}(\boldsymbol{s}_i; \boldsymbol{z}_i)$ can be represented in the form of

$$p_{\text{fp\&fn}}(\boldsymbol{s}_i; \boldsymbol{z}_i) = \prod_{j=1}^{l} \left\{ (1 - p_{\text{fp}})^{(1-s_{i,j})} p_{\text{fp}}^{s_{i,j}(1-z_{i,j})} \cdot (1 - p_{\text{d}})^{(1-s_{i,j})z_{i,j}} (p_{\text{fp}} + p_{\text{d}} - p_{\text{fp}} p_{\text{d}})^{s_{i,j} z_{i,j}} \right\}. \tag{2}$$

2.4 Our Definition of the Ordered Restriction Map Alignment Problem

Putting the models of the previous two subsections together, and taking the assumption of the probabilistic independency between sizing errors and false positives or negatives into account, the probability we get $\boldsymbol{s}_i$ from $\boldsymbol{S}$ is, in the absence of orientation errors,

$$\Pr(\boldsymbol{s}_i \mid \boldsymbol{S}) = \sum_{\boldsymbol{z}_i \in \mathcal{S}(l,r)} \left\{ p_{\text{sizing\_error}}(\boldsymbol{z}_i; \boldsymbol{S}) \cdot p_{\text{fp\&fn}}(\boldsymbol{s}_i; \boldsymbol{z}_i) \right\},$$

where $\mathcal{S}(l, r)$ denotes the set of 0-1 strings of length l with just r 1's. Thus, if the orientation of the ith molecule is indicated by d_i, the probability we get the sample set $\boldsymbol{s}_1, \cdots, \boldsymbol{s}_n$ is given by

$$\Pr(\boldsymbol{s}_1, \cdots, \boldsymbol{s}_n \mid \boldsymbol{S}) = \prod_{i=1}^{n} \left\{ \Pr(\boldsymbol{s}_i \mid \boldsymbol{S})^{1-d_i} \Pr(\overline{\boldsymbol{s}}_i \mid \boldsymbol{S})^{d_i} \right\}. \tag{3}$$

We regard (3) as a likelihood function L of $\boldsymbol{\Psi} = (\boldsymbol{\theta}^T, \boldsymbol{\sigma}^T, p_{\mathrm{d}}, p_{\mathrm{fp}}, \boldsymbol{d}^T)$, where $\boldsymbol{\theta} = (\theta_1, \cdots, \theta_r)^T, \boldsymbol{\sigma} = (\sigma_1^2, \cdots, \sigma_r^2)^T, \boldsymbol{d} = (d_1, \cdots, d_n)^T$. Then the problem is to find the estimates $\boldsymbol{\theta}^T, \boldsymbol{\sigma}^T, p_{\mathrm{d}}, p_{\mathrm{fp}}$ and 0-1 assignment of each d_i that maximize the likelihood function $L(\boldsymbol{\Psi})$ for given sample set of $\boldsymbol{s}_1, \cdots, \boldsymbol{s}_n$.

3 Local Search Based on the EM Algorithm

Here we present our local search algorithm for the problem defined in the last section. First we consider an algorithm for the simplified problem with no orientation errors. Then we extend the algorithm to a general case of orientation errors to occur.

3.1 Cases with Known Orientation

In this subsection we assume that there are no orientation errors to occur.

In this case our likelihood function to be maximized is in the form of

$$L(\boldsymbol{\Psi}) = \prod_{i=1}^{n} \Pr(\boldsymbol{s}_i \mid \boldsymbol{S}) = \prod_{i=1}^{n} \left[\sum_{\boldsymbol{z}_i \in \mathcal{S}(l,r)} \{ p_{\text{sizing\_error}}(\boldsymbol{z}_i; \boldsymbol{S}) \cdot p_{\text{fp\&fn}}(\boldsymbol{s}_i; \boldsymbol{z}_i) \} \right], \tag{4}$$

where $\boldsymbol{\Psi} = (\boldsymbol{\theta}^T, \boldsymbol{\sigma}^T, p_{\mathrm{d}}, p_{\mathrm{fp}})^T$.

We consider applying the EM algorithm [5] to this model. The EM algorithm is a broadly applicable approach to the iterative computation of maximum likelihood estimates. In order to simplify the complicated log-likelihood function, the EM algorithm assumes the existence of some additional missing data, which may be really missing, or actually unobservable. On each iteration of the EM algorithm, the *expectation step (E-step)* estimates values of missing data by utilizing conditional expectations, and the *maximization step (M-step)* maximizes the log-likelihood function using these estimated values for missing data. More detail descriptions about the EM algorithm are, for example, in [10].

In our case, it is quite natural to regard every $\boldsymbol{z}_i$, the ith molecule in an imaginary state with only sizing errors to have occurred, as unobservable missing data in the EM algorithm. The complete-data likelihood function of the EM algorithm in this case is

$$L_c(\boldsymbol{\Psi}) = \prod_{i=1}^{n} \{ p_{\text{sizing\_error}}(\boldsymbol{z}_i; \boldsymbol{S}) \cdot p_{\text{fp\&fn}}(\boldsymbol{s}_i; \boldsymbol{z}_i) \}. \tag{5}$$

Thus the log likelihood function $l_c(\boldsymbol{\Psi})$ for the complete-data is given by

$$l_c(\boldsymbol{\Psi}) = \log L_c(\boldsymbol{\Psi}) = \sum_{i=1}^{n} \log p_{\text{sizing\_error}}(\boldsymbol{z}_i; \boldsymbol{S}) + \sum_{i=1}^{n} \log p_{\text{fp\&fn}}(\boldsymbol{s}_i; \boldsymbol{z}_i). \tag{6}$$

Then Q function in the EM algorithm can be computed by taking conditional expectation given $\boldsymbol{s}_1, \cdots, \boldsymbol{s}_n$ of the complete-data log likelihood function $l_c(\boldsymbol{\Psi})$, using current estimates for parameters $\boldsymbol{\Psi}$. In the $(t+1)$th iteration of the EM algorithm, using $\boldsymbol{\Psi}^{(t)}$, estimates for $\boldsymbol{\Psi}$ in the tth iteration, as current estimates,

$$
\begin{aligned}
Q(\boldsymbol{\Psi};\boldsymbol{\Psi}^{(t)}) &= E_{\boldsymbol{\Psi}^{(t)}}(l_c(\boldsymbol{\Psi}) \mid \boldsymbol{s}_1,\cdots,\boldsymbol{s}_n) \\
&= -\sum_{i=1}^{n}\sum_{k=1}^{r}\frac{E_{\boldsymbol{\Psi}^{(t)}}(\tilde{\theta}_{i,k}^2)}{2\sigma_k^2} + \sum_{i=1}^{n}\sum_{k=1}^{r}\frac{\theta_k E_{\boldsymbol{\Psi}^{(t)}}(\tilde{\theta}_{i,k})}{\sigma_k^2} - n\sum_{k=1}^{r}\frac{\theta_k^2}{2\sigma_k^2} - \frac{n}{2}\sum_{k=1}^{r}\log(2\pi\sigma_k^2) \\
&\quad + \sum_{i=1}^{n}\sum_{j=1}^{l}(1-s_{i,j})\log(1-p_{\mathrm{fp}}) + \sum_{i=1}^{n}\sum_{j=1}^{l}\left\{1-E_{\boldsymbol{\Psi}^{(t)}}(z_{i,j})\right\}s_{i,j}\log p_{\mathrm{fp}} \\
&\quad + \sum_{i=1}^{n}\sum_{j=1}^{l}E_{\boldsymbol{\Psi}^{(t)}}(z_{i,j})(1-s_{i,j})\log(1-p_{\mathrm{d}}) + \sum_{i=1}^{n}\sum_{j=1}^{l}E_{\boldsymbol{\Psi}^{(t)}}(z_{i,j})s_{i,j}\log(p_{\mathrm{fp}}+p_{\mathrm{d}}-p_{\mathrm{fp}}p_{\mathrm{d}}).
\end{aligned}
$$

In the above equation, and in the rest of this paper, for simplicity, we use notations $E_{\boldsymbol{\Psi}^{(t)}}(\tilde{\theta}_{i,k})$, $E_{\boldsymbol{\Psi}^{(t)}}(\tilde{\theta}_{i,k}^2)$, $E_{\boldsymbol{\Psi}^{(t)}}(z_{i,j})$ instead of $E_{\boldsymbol{\Psi}^{(t)}}(\tilde{\theta}_{i,k} \mid \boldsymbol{s}_1,\cdots,\boldsymbol{s}_n)$, $E_{\boldsymbol{\Psi}^{(t)}}(\tilde{\theta}_{i,k}^2 \mid \boldsymbol{s}_1,\cdots,\boldsymbol{s}_n)$, $E_{\boldsymbol{\Psi}^{(t)}}(z_{i,j} \mid \boldsymbol{s}_1,\cdots,\boldsymbol{s}_n)$, respectively.

Hence, if we can evaluate each conditional expectation $E_{\boldsymbol{\Psi}^{(t)}}(\tilde{\theta}_{i,k})$, $E_{\boldsymbol{\Psi}^{(t)}}(\tilde{\theta}_{i,k}^2)$, $E_{\boldsymbol{\Psi}^{(t)}}(z_{i,j})$, the M-step on the $(t+1)$ the iteration can be fulfilled as follows.

By computing partial derivative with respect to each θ_k, σ_k^2, and solving simultaneous partial differential equations, we can maximize Q with respect to each θ_k, σ_k^2, respectively, by following updates:

$$
\theta_k^{(t+1)} := \frac{1}{n}\sum_{i=1}^{n}E_{\boldsymbol{\Psi}^{(t)}}(\tilde{\theta}_{i,k}), \quad \sigma_k^{2\,(t+1)} := \frac{1}{n}\sum_{i=1}^{n}E_{\boldsymbol{\Psi}^{(t)}}(\tilde{\theta}_{i,k}^2) - \theta_k^{(t+1)\,2}, \tag{7}
$$

for all $k = 1,\cdots,r$. Similarly, as for $p_{\mathrm{d}}, p_{\mathrm{fp}}$, by computing partial derivative with respect to $p_{\mathrm{d}}, p_{\mathrm{fp}}$, and solving simultaneous partial differential equations, we can maximize Q with respect to $p_{\mathrm{d}}, p_{\mathrm{fp}}$, respectively, by following updates:

$$
p_{\mathrm{fp}}^{(t+1)} := \frac{\displaystyle\sum_{i=1}^{n}\sum_{j=1}^{l}\left(s_{i,j} - E_{\boldsymbol{\Psi}^{(t)}}(z_{i,j})s_{i,j}\right)}{\displaystyle nl - \sum_{i=1}^{n}\sum_{j=1}^{l}E_{\boldsymbol{\Psi}^{(t)}}(z_{i,j})}, \quad p_{\mathrm{d}}^{(t+1)} := \frac{\displaystyle\sum_{i=1}^{n}\sum_{j=1}^{l}E_{\boldsymbol{\Psi}^{(t)}}(z_{i,j})s_{i,j} - p_{\mathrm{fp}}^{(t+1)}\sum_{i=1}^{n}\sum_{j=1}^{l}E_{\boldsymbol{\Psi}^{(t)}}(z_{i,j})}{\displaystyle (1-p_{\mathrm{fp}}^{(t+1)})\sum_{i=1}^{n}\sum_{j=1}^{l}E_{\boldsymbol{\Psi}^{(t)}}(z_{i,j})}. \tag{8}
$$

Finally we consider the way of computing efficiently each conditional expectation $E_{\boldsymbol{\Psi}^{(t)}}(\tilde{\theta}_{i,k})$, $E_{\boldsymbol{\Psi}^{(t)}}(\tilde{\theta}_{i,k}^2)$, $E_{\boldsymbol{\Psi}^{(t)}}(z_{i,j})$. Taking advantage of our discretized formulation, we can utilize Dynamic Programming (DP) and compute values of these conditional expectations efficiently.

Notice that the probability we get $\boldsymbol{s}_i$ from $\boldsymbol{S}$ can be rewritten in the form of

$$
\begin{aligned}
\Pr(\boldsymbol{s}_i \mid \boldsymbol{S}) &= \sum_{1\le\tilde{\theta}_{i,1}<\cdots<\tilde{\theta}_{i,r}\le l}\left\{\prod_{m=1}^{r}g(\tilde{\theta}_{i,m};\theta_m,\sigma_m^2)\cdot\prod_{j=1}^{l}p_{\mathrm{bit\_error}}(s_{i,j};0,p_{\mathrm{d}},p_{\mathrm{fp}})\right. \\
&\qquad \left.\cdot\prod_{m=1}^{r}\frac{p_{\mathrm{bit\_error}}(s_{i,\tilde{\theta}_{i,m}};1,p_{\mathrm{d}},p_{\mathrm{fp}})}{p_{\mathrm{bit\_error}}(s_{i,\tilde{\theta}_{i,m}};0,p_{\mathrm{d}},p_{\mathrm{fp}})}\right\}.
\end{aligned}
$$

Let $\xi_{i,1}(x,k), \xi_{i,2}(x,k)$ be defined as follows:

$$
\begin{aligned}
\xi_{i,1}(x,k) &= \sum_{1\le\tilde{\theta}_{i,1}<\cdots<\tilde{\theta}_{i,k}\le x}\left\{\prod_{m=1}^{k}g(\tilde{\theta}_{i,m};\theta_m,\sigma_m^2)\cdot\prod_{j=1}^{x}p_{\mathrm{bit\_error}}(s_{i,j};0,p_{\mathrm{d}},p_{\mathrm{fp}})\right. \\
&\qquad \left.\cdot\prod_{m=1}^{k}\frac{p_{\mathrm{bit\_error}}(s_{i,\tilde{\theta}_{i,m}};1,p_{\mathrm{d}},p_{\mathrm{fp}})}{p_{\mathrm{bit\_error}}(s_{i,\tilde{\theta}_{i,m}};0,p_{\mathrm{d}},p_{\mathrm{fp}})}\right\},
\end{aligned} \tag{9}
$$

$$\xi_{i,2}(x,k) = \sum_{x \le \tilde{\theta}_{i,k} < \cdots < \tilde{\theta}_{i,r} \le l} \left\{ \prod_{m=k}^{r} g(\tilde{\theta}_{i,m}; \theta_m, \sigma_m^2) \cdot \prod_{j=x}^{l} p_{\text{bit\_error}}(s_{i,j}; 0, p_{\text{d}}, p_{\text{fp}}) \right.$$
$$\left. \cdot \prod_{m=k}^{r} \frac{p_{\text{bit\_error}}(s_{i,\tilde{\theta}_{i,m}}; 1, p_{\text{d}}, p_{\text{fp}})}{p_{\text{bit\_error}}(s_{i,\tilde{\theta}_{i,m}}; 0, p_{\text{d}}, p_{\text{fp}})} \right\}. \quad (10)$$

Note that $\Pr(\boldsymbol{s}_i \mid \boldsymbol{S}) = \xi_{i,1}(l,r) = \xi_{i,2}(1,1)$.

Then the conditional probability given $\boldsymbol{S}, \boldsymbol{s}_1, \cdots, \boldsymbol{s}_n$ that $\tilde{\theta}_{i,k} = x$ is

$$\Pr(\tilde{\theta}_{i,k} = x \mid \boldsymbol{S}, \boldsymbol{s}_1, \cdots, \boldsymbol{s}_n) = \Pr(\tilde{\theta}_{i,k} = x \mid \boldsymbol{S}, \boldsymbol{s}_i) = \frac{\Pr(\boldsymbol{s}_i \cap \{\tilde{\theta}_{i,k} = x\} \mid \boldsymbol{S})}{\Pr(\boldsymbol{s}_i \mid \boldsymbol{S})}$$
$$= \frac{1}{\xi_{i,1}(l,r)} \cdot \xi_{i,1}(x-1, k-1) \cdot g(x; \theta_k, \sigma_k^2) p_{\text{bit\_error}}(s_{i,x}; 1, p_{\text{d}}, p_{\text{fp}}) \cdot \xi_{i,2}(x+1, k+1). \quad (11)$$

The main advantage of using such $\xi_{i,1}(x,k), \xi_{i,2}(x,k)$ is that we can represent these $\xi_{i,1}(x,k)$, $\xi_{i,2}(x,k)$ in the following inductive forms:

$$\xi_{i,1}(x,k) = \xi_{i,1}(x-1,k-1) \cdot g(x; \theta_k, \sigma_k^2) p_{\text{bit\_error}}(s_{i,x}; 1, p_{\text{d}}, p_{\text{fp}}) + \xi_{i,1}(x-1,k) \cdot p_{\text{bit\_error}}(s_{i,x}; 0, p_{\text{d}}, p_{\text{fp}}),$$

$$\xi_{i,2}(x,k) = \xi_{i,2}(x+1,k+1) \cdot g(x; \theta_k, \sigma_k^2) p_{\text{bit\_error}}(s_{i,x}; 1, p_{\text{d}}, p_{\text{fp}}) + \xi_{i,2}(x+1,k) \cdot p_{\text{bit\_error}}(s_{i,x}; 0, p_{\text{d}}, p_{\text{fp}}).$$

Thus, using Dynamic Programming (DP), we can compute all values of $\xi_{i,1}(x,k), \xi_{i,2}(x,k)$ in $O(lr)$ time and space ($x = 1, \cdots, l, k = 1, \cdots, r$). Hence we can compute the conditional probability given $\boldsymbol{S}, \boldsymbol{s}_1, \cdots, \boldsymbol{s}_n$ that $\tilde{\theta}_{i,k} = x$ from (11). Therefore the conditional expectations can be computed as follows ($i = 1, \cdots, n, j = 1, \cdots, l, k = 1, \cdots, r$):

$$E_{\boldsymbol{\Psi}^{(t)}}(\tilde{\theta}_{i,k} \mid \boldsymbol{s}_1, \cdots, \boldsymbol{s}_n) = \sum_{x=k}^{l-r+k} \left\{ x \cdot \Pr(\tilde{\theta}_{i,k} = x \mid \boldsymbol{s}_1, \cdots, \boldsymbol{s}_n) \right\}, \quad (12)$$

$$E_{\boldsymbol{\Psi}^{(t)}}(\tilde{\theta}_{i,k}^2 \mid \boldsymbol{s}_1, \cdots, \boldsymbol{s}_n) = \sum_{x=k}^{l-r+k} \left\{ x^2 \cdot \Pr(\tilde{\theta}_{i,k} = x \mid \boldsymbol{s}_1, \cdots, \boldsymbol{s}_n) \right\}, \quad (13)$$

$$E_{\boldsymbol{\Psi}^{(t)}}(z_{i,j} \mid \boldsymbol{s}_1, \cdots, \boldsymbol{s}_n) = \sum_{k=1}^{r} \Pr(\tilde{\theta}_{i,k} = j \mid \boldsymbol{s}_1, \cdots, \boldsymbol{s}_n). \quad (14)$$

Now we get the following proposition:

Proposition 1 *All the values of conditional expectations* $E_{\boldsymbol{\Psi}^{(t)}}(\tilde{\theta}_{i,k})$, $E_{\boldsymbol{\Psi}^{(t)}}(\tilde{\theta}_{i,k}^2)$, $E_{\boldsymbol{\Psi}^{(t)}}(z_{i,j})$ ($i = 1, \cdots, n, j = 1, \cdots, l, k = 1, \cdots, r$) *can be computed in* $O(nlr)$ *time.*

3.2 Cases with Unknown Orientation

Here we extend our algorithm to a general case that there happens orientation errors.

In this case the log likelihood function we want to maximize is

$$l(\boldsymbol{\Psi}) = \sum_{i=1}^{n} (1 - d_i) \log \Pr(\boldsymbol{s}_i \mid \boldsymbol{S}) + \sum_{i=1}^{n} d_i \log \Pr(\overline{\boldsymbol{s}}_i \mid \boldsymbol{S}). \quad (15)$$

We regard the problem of determining orientation of each molecule as a classification problem with two mixture components, and utilize the concept of *Classification EM (CEM)* algorithm [3].

At each iteration, we first determine orientation of every molecule, by assigning 1 to each $d_i^{(t)}$ if $\Pr(\boldsymbol{s}_i \mid \boldsymbol{S}^{(t)}) < \Pr(\overline{\boldsymbol{s}}_i \mid \boldsymbol{S}^{(t)})$ or 0 otherwise. Then we fix orientations and fulfill one iteration of the EM algorithm in the previous subsection. In this case we replace each $s_{i,j}$ with $M(\boldsymbol{d}^{(t)})_{ij}$ in (7), (8). The whole algorithm goes as follows:

Algorithm 1

1. *Compute initial solutions* $\boldsymbol{S}^{(0)} = \boldsymbol{\theta}^{(0)}, \boldsymbol{\sigma}^{(0)}, p_{\mathrm{d}}^{(0)}, p_{\mathrm{fp}}^{(0)}$ *for* $\boldsymbol{S} = \boldsymbol{\theta}, \boldsymbol{\sigma}, p_{\mathrm{d}}, p_{\mathrm{fp}}$.
2. *Start from* $t = 0$ *and repeat the following steps until convergence with sufficient accuracy:*
   - 2.1 *Assign 0 or 1 to each* $d_i^{(t)}$ $(i = 1, \cdots, n)$.
   - 2.2 *Compute the conditional expectations* $E_{\boldsymbol{\Psi}^{(t)}}(\tilde{\theta}_{i,k})$, $E_{\boldsymbol{\Psi}^{(t)}}(\tilde{\theta}_{i,k}^2)$, $E_{\boldsymbol{\Psi}^{(t)}}(z_{i,j})$ *given* $\boldsymbol{S}^{(t)}$, $M(\boldsymbol{d}^{(t)})$ *using DP* $(i = 1, \cdots, n, j = 1, \cdots, l, k = 1, \cdots, r)$.
   - 2.3 *Compute the* $(t+1)$*th estimate for each* θ_k, σ_k^2 $(k = 1, \cdots, r)$.
   - 2.4 *Compute the* $(t+1)$*th estimate for* $p_{\mathrm{d}}, p_{\mathrm{fp}}$.
3. *If* $\sum_{i=1}^n d_i^{(t)} > \frac{n}{2}$, *reverse the estimated map.*

4 An Efficient Heuristic Algorithm for Initial Solution

Here we present an efficient heuristic algorithm to obtain initial solution of good quality. First we explain how to determine initial $\boldsymbol{\theta}$, locations of restriction sites, with given $r, \sigma, p_{\mathrm{d}}, p_{\mathrm{fp}}$, where we assume that $\sigma = \sigma_1 = \cdots = \sigma_r$ in given initial $\boldsymbol{\sigma}$. A method for determining initial $r, \sigma, p_{\mathrm{d}}, p_{\mathrm{fp}}$ will be presented later.

We regard the number of 1's on the jth position of the data, that is, $\sum_{i=1}^n s_{i,j}$, as a frequency on the jth position, and consider approximating distribution of this frequency by that of expected frequency from some mixture of normal distributions. However, due to the existence of the orientation errors, the distribution of 1's in the data set cannot be approximated directly by that of the mixture of normal distributions. To overcome this, we consider $\sum_{i=1}^n (s_{i,j} + \bar{s}_{i,j})$ for $j = 1, \cdots, \lceil l/2 \rceil$, instead of $\sum_{i=1}^n s_{i,j}$ for $j = 1, \cdots, l$, and enumerate r pairs $j, \bar{j} = l - j + 1$, at first. Then next we choose exactly one of $j, \bar{j}$ as an initial restriction site for each candidate enumerated at the first step. For convenience, we assume that l is even. The precise definition of the algorithm based on the above idea is as follows:

Algorithm 2

1. *Compute* $c_j = \sum_{i=1}^n (s_{i,j} + s_{i,l-j+1})$ *for* $j = 1, \cdots, l/2$.
2. *Repeat the following steps for* r *times:*
   - 2.1 *Find* $j_{\max} = \arg\max_j \{c_j\}$, *and store the pair of* $j_{\max}$ *and the corresponding* $\bar{j}_{\max}$ *as a candidate pair for the restriction site. If* $j_{\max}$ *has been already stored as a candidate in the earlier step, we check* $j_{\max} \pm 1$ *and choose* $j_{\max} - 1$ *or* $j_{\max} + 1$ *with larger* c *value than the other. If both of* $j_{\max} \pm 1$ *have also been already stored, we check* $j_{\max} \pm 2$, *and so on.*
   - 2.2 *Update* c *values as* $c_j := c_j - n(1 - p_{\mathrm{fp}}) p_{\mathrm{d}} g(j; j_{\max}, \sigma^2)$ *for* $j = 1, \cdots, l/2$.
3. *Choose* j *or* $\bar{j}$ *as an initial restriction site for* r *pairs of sites enumerated in step 2. There are* 2^{r-1} *possible combinations of the restriction sites, since we can fix, without losing generality, the choice of* j *or* $\bar{j}$ *for one of the* r *candidate pairs. We choose the combination which maximize* $\prod_{i=1}^n \max\{\Pr(\boldsymbol{s}_i \mid \boldsymbol{S}), \Pr(\overline{\boldsymbol{s}}_i \mid \boldsymbol{S})\}$.

Here we present how to determine initial $r, \sigma, p_{\mathrm{d}}, p_{\mathrm{fp}}$ from given data set $\boldsymbol{s}_1, \cdots, \boldsymbol{s}_n$. Let N be the total number of 1's appearing in $\boldsymbol{s}_1, \cdots, \boldsymbol{s}_n$, or, $N = \sum_{i=1}^{n} \sum_{j=1}^{l} s_{i,j}$ equivalently. If initial $r, \sigma, p_{\mathrm{d}}$ are given,

$$p_{\mathrm{fp}} = \frac{N - nrp_{\mathrm{d}}}{nl - nrp_{\mathrm{d}}} \tag{16}$$

is the unbiased estimator for p_{fp}. Thus we have only to determine initial $r, \sigma, p_{\mathrm{d}}$.

Our method to determine initial $r, \sigma, p_{\mathrm{d}}$ is based on multi start heuristics. We test Algorithm 2 for all possible sets of $r, \sigma, p_{\mathrm{d}}$, such that

$$r = i, \sigma = \{0.01 + 0.005(j-1)\}l, p_{\mathrm{d}} = 0.3 + 0.025(k-1) \ (i = 1, \cdots, 10, j = 1, \cdots, 5, k = 1, \cdots, 17).$$

Thus we test 850 possible sets for $r = 1, \cdots, 10$, $0.01l \leq \sigma \leq 0.03l$, $0.3 \leq p_{\mathrm{d}} \leq 0.7$. We take as initial values the set of $r, \sigma, p_{\mathrm{d}}$ that maximizes the probability in step 3 of Algorithm 2 among all of these 850 possible start sets.

In practice, computing probability in step 3 contains $O(nlr)$ times of computation of function $g(x; \mu, \sigma^2)$, which is too expensive to repeat Algorithm 2 many (850) times. Therefore, we compute probability in step 3 of Algorithm 2 approximately in our program. Also we found that it is better to introduce some penalty term when computing the probability in step 3 of Algorithm 2, in order to avoid redundant expression by overestimating r. Details are in [8] and we omit explanations here.

5 Experimental Results

Here we show our experimental results. Due to the constraints on disclosure, we cannot present the results with real data sets. Hence we show the results with our simulated data sets. Although we have experimented with several sets of simulated data, we show only one of them here, since we have not enough space. Our other results are available in [8].

In our result below, the location of each restriction site is represented by the relative position of the unit interval $(0, 1)$, and standard deviation σ_k associated to each restriction site is hence represented by the value of this measurement. Computations were done on Sun UltraSPARC-II, 360 MHz workstation with 2048 MB memory.

In our experiment, we set $l = 200$. We show a result with a data set of $r = 7$, $p_{\mathrm{d}} = 0.3125$, $p_{\mathrm{fp}} = 0.03$, $\sigma = \sigma_1 = \cdots = \sigma_7 = 1/60$. The expected number of false cuts per molecule is 6, and 45 % of molecules are reversed. In the standard case for real data sets, we believe, $p_{\mathrm{d}} \approx 0.5$ and the expected number of false cuts per molecule is about 1. Thus this example will be quite more difficult than the actual cases, because of low digestion rate and a good many false positive noises. Locations of restriction sites are

$$\boldsymbol{\theta} = (0.15000, 0.30000, 0.40000, 0.55000, 0.80000, 0.85000, 0.90000)^T.$$

We used $n = 100$ molecules, and obtained an initial solution of

$$\begin{aligned}
&\boldsymbol{\theta} = (0.09750, 0.16250, 0.21750, 0.45250, 0.59250, 0.67750, 0.87750)^T, \\
&\boldsymbol{\sigma} = (0.01000, 0.01000, 0.01000, 0.01000, 0.01000, 0.01000, 0.01000)^T, \\
&p_{\mathrm{d}} = 0.30000, \quad p_{\mathrm{fp}} = 0.03052,
\end{aligned}$$

after about 4 minutes computation. Then, by our local search algorithm, we obtained a true map

$$\begin{aligned}
&\boldsymbol{\theta} = (0.12865, 0.28951, 0.40318, 0.54217, 0.78121, 0.84100, 0.90301)^T, \\
&\boldsymbol{\sigma} = (0.01078, 0.02373, 0.01786, 0.00796, 0.01298, 0.00983, 0.00959)^T, \\
&p_{\mathrm{d}} = 0.35590, \quad p_{\mathrm{fp}} = 0.02860,
\end{aligned}$$

after 74 iterations, and the computational time for local search was about 1.5 minutes. The stopping criterion is $\max_k |\theta_k^{(t+1)} - \theta_k^{(t)}| \leq 10^{-2}/l = 1/20000$.

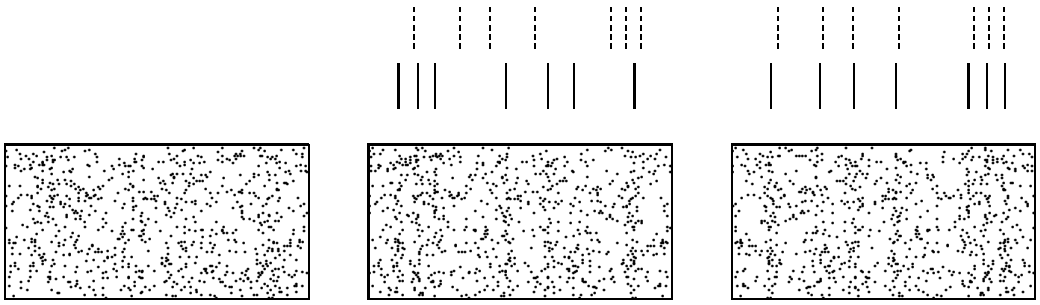

Figure 1: Data set, initial solution, and final solution of simulated data with $n = 100$, $r = 7$, $p_{\mathrm{d}} = 0.3125$, $p_{\mathrm{fp}} = 0.03$, $\sigma = \sigma_1 = \cdots = \sigma_7 = 1/60$.

In this case our heuristic algorithm for initial solution estimated the reversed map. However, the local search algorithm could find the actual map owing to step 3 of Algorithm 1.

For a measure of accuracy of the solution map in comparison with the true map, we use the *Mislocation Measure* percentage (MMP) error defined by Geiger and Parida [6]. The MMP error is defined by, if the estimated number of restriction sites is correct, the percentage with respect to l of the maximum of the absolute distance of a cut site in the true map from its corresponding cut in the computed map. In this case the MMP error is about 2.2 % (with the case of θ_1).

Fig. 1 shows the data set, the initial solution, and the final solution from left to right, respectively. Each row of the image is a molecule with each dot indicating an observed cut site. In the images of the initial solution and final solution, the molecules are oriented according to the solution computed by the algorithm. The dash lines above each map show the locations of the actual restriction sites, while the solid lines show the locations of the estimated restriction sites.

In other experiments, our algorithm sometimes overestimates r, the number of restriction sites. However, our algorithm found all true restriction sites even for such cases, although it also answered some extra cut sites. And it is often the case that this disadvantage of overestimating r can be overcome by increasing n, the number of molecules.

6 Conclusions

In this paper we gave a statistical model for the ordered restriction map alignment problem, and considered algorithms based on our model.

We formulated the problem in a statistical way, incorporating the merit of combinatorial approach by using the discretized model in order to simplify the model. We defined our statistical model to deal with three types of the error factors, sizing errors, false negatives or positives, and orientation errors. Then we applied the EM algorithm for removing the noises of first two types of error factors, assuming imaginary state in which only sizing errors take place. For orientation errors, we utilized the concept of two-clustering.

Since the EM-type algorithms are local search algorithms, and their accuracy greatly depends on the initial solution, we also proposed an efficient heuristic algorithm for finding initial solution with a good quality.

In our experiments, our algorithm worked well for a data set which we believe more difficult than the actual cases.

As for future works for this problem, we should deal with other types of error factors such as the existence of spurious molecules or missing fragments. We should also examine more theoretically and experimentally the property of the penalty term used in our heuristic algorithm for initial solution.

Acknowledgement

This work was supported in part by the Grant-in-Aid for Scientific Research on Priority Areas, "Genome Science" from the Ministry of Education, Science, Sports and Culture of Japan.

References

[1] Anantharaman, T.S., Mishra, B., and Schwartz, D.C., Genomics via optical mapping II: ordered restriction maps, *J. Computational Biology*, 4(2):91–118, 1997.

[2] Cai, W., Aburatani, H., Housman, D., Wang, Y., and Schwartz, D.C., Ordered restriction endonuclease maps of yeast artificial chromosomes created by optical mapping on surfaces, *Proc. Natl. Acad. Sci. USA*, 92:5164–5168, 1995.

[3] Celeux, G. and Govaert, G., A classification EM algorithm for clustering and two stochastic versions, *J. Computational Statistics & Data Analysis*, 14:315–332, 1992.

[4] Dančík, V. and Waterman, M.S., Simple maximum likelihood methods for the optical mapping problem, *Genome Informatics 1997*, 1–8, 1997.

[5] Dempster, A.P., Laird, N.M., and Rubin, D.B., Maximum likelihood from incomplete data via the EM algorithm (with discussion), *J. the Royal Statistical Society B*, 39:1–38, 1977.

[6] Geiger, D. and Parida, L., Mass estimation of DNA molecules & extraction of ordered restriction maps in optical mapping imagery, *Algorithmica*, Special Issue on Computational Biology, 1999 (to appear).

[7] Karp, R.M. and Shamir, R., Algorithms for optical mapping, *Proc. of 2nd ACM Conference on Computational Molecular Biology*, 117–124, 1998.

[8] Kobayashi, H., Investigation of Algorithms for Biological Alignment Problems, A Master's Thesis, Department of Information Science, University of Tokyo, Mar. 1999.

[9] Lee, J.K., Dančík, V., and Waterman, M.S., Estimation for restriction sites observed by optical mapping using reversible-jump Markov chain Monte Carlo, *Proc. 2nd ACM Conference on Computational Molecular Biology*, 147–152, Mar. 1998.

[10] McLachlan, G.J. and Krishnan, T., *The EM Algorithm and Extensions*, John Wiley & Sons, New York, 1997.

[11] Meng, X., Benson, K., Chada, K., Huff, E.J., and Schwartz, D.C., Optical mapping of lambda bacteriophage clones using restriction endonucleases, *Nature Genetics*, 9:432–438, 1995.

[12] Muthukrishnan, S. and Parida, L., Towards constructing physical maps by optical mapping: an effective, simple, combinatorial approach (extended abstract), *Proc. 1st ACM Conference on Computational Molecular Biology*, 209–219, Jan. 1997.

[13] Parida, L., A uniform framework for ordered restriction map problems, *J. Computational Biology*, 5(4):725–739, 1998.

[14] Parida, L., On the approximability of physical map problems using single molecule methods, *Proc. 2nd Discrete Mathematics and Theoretical Computer Science Conference*, 1999 (to appear).

[15] Schwartz, D.C., Li, X., Hernandez, L.I., Ramnarain, S.P., Huff, E.J., and Wang, Y.K., Ordered restriction maps of *Saccharomyces cerevisiae* chromosomes constructed by optical mapping, *Science*, 262:110–114, 1993.

[16] Wang, Y.K., Huff, E.J., and Schwartz, D.C., Optical mapping of site-directed cleavages on single DNA molecules by the RecA-assisted restriction endonuclease technique, *Proc. Natl Acad. Sci. USA*, 92:165–169, 1995.