A Model and Solution to the DNA flipping string problem

Davi Geiger and Laxmi Parida
Courant Institute, Dept. of Computer Science
New York University
251 Mercer street
New York, NY 10012-1185
geiger@cs.nyu.edu
parida@cs.nyu.edu

Abstract

We consider the case where a pool of DNA molecules clones both, flipped and not-flipped, have been cut by restriction enzymes. Ideally, each clone is cut in the same positions, although in practice due to errors, this does not always happen. The computational problem is to determine where the cuts have occurred.

This is a key problem in determining the structure of the original DNA molecule.

A single molecule is represented by a string of 1's and 0's, with cuts represented by $1's$. A set of molecules clones (with errors) is observed, but the orientation/parity of each molecule is unknown. Clear is that the location of the observed cuts of one molecule are dependent on the parity: flipping the molecule would result in the cuts location, as observed, being "flipped" .

We propose a Bayesian approach to generate a posterior distribution on the cuts and parity, given the data. We first present an approximate algorithm where we attempt to divide the problem into subproblems, but it is not guaranteed to solve the problem. Then, we propose another approximate method based on a statistical framework and a mean field annealing algorithm. It computes the maximum posterior marginal (MPM estimator) and maximum aposteriori estimate (MAP estimator).

We also provide evidence that the exact solution of the problem is intractable.

1 Introduction

We start by giving a background on the origins of the problem and then give a brief description of the problem.

1.1 Background and Problem Description

Restriction enzymes *cut* a DNA molecule at certain specific base pair pattern termed *site*. A biologist obtains valuable information from knowing the order of the *sites* along with distances between each of them. This is called the *Physical Mapping Problem*.

Given different pieces of DNA molecules, the *Contig Problem* is to find the overlap between these strands. This is done by first obtaining the physical map of each molecule, and then using the physical maps to detect the overlap between the DNA molecules.

What kind of information does a biologist/chemist provide a computer scientist? Firstly, for the physical map, the computer scientist is given the results of an experiment carried out in a laboratory: hence this is technology driven. Gel Eleclotrophoresis and Optical Mapping (see [2], [9]) are two methods to provide input for the physical mapping problem.

Gel Electrophoresis is a popular method of studying the fragments of DNA molecules obtained by digesting with restriction enzymes. It gives the number of fragments of a particular size, but gives no information of the order of these segments on the DNA molecule. The physical mapping problem is to find this order.

Optical Mapping was introduced by David Schwartz (e.g., [10]) and has been a significant breakthrough in the experimental method to study the cutting of DNA molecules by restriction enzymes.

The current optical mapping technique is the second generation approach which can use DNA fragments as small as 300 base pairs[9].[1] The process fixes elongated DNA molecules onto polylysine-treated glass surfaces. The fixation conditions are carefully controlled to minimize DNA coil relaxation effects but allow enough relaxation at endonuclease cleavage sites for their detection by flourescence microscopy. [2] This surface based optical mapping approach will provide the necessary basis for the development of a fully automated approach to genomic analysis that should eliminate the need for electrophoretic techniques.

[1] The first generation approach used agarose gel and was less accurate; the second generation is combining simplicity of the procedure along with increased accuracy.

[2] Here we are talking about a balance between a molecule coiling up into a ball, which gives very little or no information about its length, or not retracting at all so as to completely hide the cutting site.

What is the advantage of optical mapping over conventional methods? It is believed that optical mapping is more suitable for automated approach than other conventional methods. The automated approach should help analyze genomes of other biologically valuable organisms.

It must be pointed out that the *digestion rate* is fairly high in the Gel Electrophoresis method but is poor in the optical method. [3] In the former it is believed to be 99% whereas in the latter it could be as low as 30%. But, recent experiments have shown that this can be improved significantly by using specially treated surfaces.

The error model in optical mapping is also a little different from that of the electrophoretic method: now there could be cuts that are optical and actually don't exist on the molecules. This is the *Type II* or *false positive* error. Due to unexplained reasons, sometimes the restriction enzymes fail to cleave at a site: this is the *Type I* or *false negative* error.

The reader may note that the contig problem is independent of the way by which the digestion is carried out: hence it is the same in both the approaches.

Let us summarize the pros and cons of the two techniques in the following table:

Techniques	The Physical Map Problem				The Contig Problem	Automation
	Digestion Rate	Error Types	Order of fragments	Computational Problems		
Gel Electrophoresis	good (99%)	Type I	unknown	hard combinatorial	same	not conducive
Optical Technique	fair (33%)	Type I & II	**known**	vision & tractable combinatorial	same (but, more reliable result)	conducive

In the table, we say that the contig problem, using the Optical technique, gives a more reliable result, since its input comes from the "better" Physical map (since the ordering is known).

1.2 Detection of single DNA molecules

In our previous work [5] and master thesis guided by one of the authors [8] we have considered algorithms to extract DNA molecules from microscopic images (in the context of the optical mapping approach). The idea was to describe DNA molecules as a smooth chain (Markov chain) of light grey values (light

[3] The efficiency with which the restriction endonucleuses digest (cut) the DNA molecule.

compared to the background) with possible gaps (due to the activity of the restriction enzymes). We have employed dynamic programming and Dijkstra algorithms to find optimal solutions to our model.

A problem arises since these molecules have inherent errors, as described before, and a statistical analysis of the results generated by the detection algorithm needs to take place. Thus, our work extends our previous work on detecting individual DNA's molecules.

1.3 The problem

We can now describe our proposed problem based on the data obtained by the molecule detection algorithm. The data $d_{ij} = 0, 1$ indicates if molecule i at the given orientation (parity) has a cut at position x_j. Thus the input data is a set of m strings ("molecules"), each one of length n with values 0's and 1's.

For example consider:

10100100010	10100100010
01000110101	10101100010
10100100010	10100100010
10100000010	10100000010
01000100101	10100100010
10000100010	10000100010
	10100100010

The original molecule is the string in the right and bottom. The input is the array on the left and the output, i.e., the flipping solution is the array on the right. The reader may verify that the array on the right is the "best" possible arrangement. Rows 2 and 5 have been flipped to obtain the alignment of 1's in columns 1, 3, 6 and 10. The original molecule, the string in the right and bottom, is obtained by maximizing the columns correlation on the right (after flipping has been solved).

Now we define a criteria for the "best" possible arrangement.

2 Formulation of the Computational Problem

We introduce the binary processes $\{P_i = -1, 1\ ; i = 1, ..., m\}$ and $\{Y_j = 0, 1\ ; j = 1, ..., n\}$. $P_i = -1$ indicates that molecule i has parity -1 and needs to be flipped to have the same orientation as everyone else. When flipping a molecule the location of the cuts change to their complementary position. $P_i = 1$ keeps the orientation as it is. $Y_j = 1$ is a decision that a cut occur at position x_j of the "true" molecule (the one we started to copy from).

We define a probability measure on the data given the variables $\{P_i\}, \{Y_j\}$, assuming n is even[4],

$$P(\{d_{ij}\}|\{P_i\},\{Y_j\}) = \frac{1}{Z} e^{-\beta E(\{Y_j,P_i\})} = \frac{1}{Z} \prod_{i=1}^{m} \prod_{j=1}^{n/2} e^{-\beta E_{ij}(Y_j,P_i)}$$

$$= \frac{1}{Z} \prod_{i=1}^{m} \prod_{j=1}^{n/2} e^{\beta \left\{ Y_j \frac{1}{2}[(1+P_i)d_{ij}+(1-P_i)d_{i,n-j+1}]+(1-Y_j)\frac{1}{2}[(1+P_i)(1-d_{ij})+(1-P_i)(1-d_{i,n-j+1})] \right\}} ,$$

where β is a parameter that does not alter the ordering of the distribution, but reflects the uncertainty in the model. In physics it plays the role of the inverse of the temperature of the system. Z is the normalization constant known as the partition function. This model encourages the detection of cuts, $Y_j = 1$, when "cut data" is observed at either (i) location x_j if the parity is $P_i = 1$ or (ii) location x_{n-j+1} if the parity is reversed, $P_i = -1$. Moreover, it inhibits the detection of cuts, $Y_j = 0$, when "cut data" is not observed, again, at either (i) location x_j if the parity is $P_i = 1$ or (ii) location x_{n-j+1} if the parity is reversed, $P_i = -1$.

In the absense of a prior distribution on $\{P_i, Y_j\}$ we consider a maximum likelihood approach, i.e., we assume a uniform distribution on $\{P_i\}$ and $\{Y_j\}$ to obtain

$$P(\{P_i\},\{Y_j\}|\{d_{ij}\}) = P(\{d_{ij}\}|\{P_i\},\{Y_j\}) \,.$$

By changing variables, $X_i = \frac{1}{2}(1 - P_i)$, i.e., $X_i = 1$ if a flip occurs and $X_i = 0$ if it does not, the energy reduces to

$$\begin{aligned}
E(Y,X) &= \sum_{i=1}^{m}\sum_{j=1}^{n/2} E_{ij}(Y_j,X_i) \\
&= -\sum_{i=1}^{m}\sum_{j=1}^{n/2} Y_j\left[(1-X_i)d_{ij} + X_i d_{i,n-j+1}\right] \\
&\qquad + (1-Y_j)[(1-X_i)(1-d_{ij}) + X_i(1-d_{i,n-j+1})] \\
&= -\sum_{i=1}^{m}\sum_{j=1}^{n/2}[\, 2Y_jX_i(d_{i,n-j+1} - d_{i,j}) + X_i(d_{ij} - d_{i,n-j+1}) \\
&\qquad + Y_j(2d_{i,j} - 1) + (1 - d_{i,j})\,] \\
&= A - \sum_{i=1}^{m}\sum_{j=1}^{n/2} \tau_{ij}^{XY} Y_j X_i - \sum_{i=1}^{m} \tau_i^X X_i - \sum_{j=1}^{n/2} \tau_j^Y Y_j\,] , \qquad (1)
\end{aligned}$$

[4] If n is odd, we just remove the $\frac{n+1}{2}$-th column. Note that this does not affect the flipping problem.

where $A = (1 - d_{i,j})$, and

$$\left\{ \begin{array}{ccc} \tau_i^X & = & \sum_{j=1}^{n/2}(d_{i,j} - d_{i,n-j+1}) \\ \tau_j^Y & = & \sum_{i=1}^{m}(2d_{i,j} - 1) \\ \tau_{ij}^{XY} & = & 2(d_{i,n-j+1} - d_{i,j}) \end{array} \right. .$$

Let us look at the coefficients of the various terms and their properties:

terms	coefficients	observations
X_i	$\tau_i^X = \sum_j^{n/2}(d_{ij} - d_{i,n-j+1})$	• $\tau_i^X = -\frac{\sum_j \tau_{ij}^{XY}}{2}$ • $-n/2 \leq \tau_i^X \leq n/2$ (the data can be simply altered so that $0 \leq \tau_i^X \leq n/2$)
Y_j	$\tau_j^Y = \sum_i^m(2d_{ij} - 1)$	• $-m \leq \tau_j^Y \leq m$
X_iX_k	0	
Y_jY_k	0	
X_iY_j	$\tau_{ij}^{XY} = 2(d_{i,n-j+1} - d_{ij})$	$\tau_{ij}^{XY} \in \{-2, 0, 2\}$

Noting that $X_i^2 = X_i$ for $X_i = 0, 1$, we can also write this energy in terms of a quadratic form

$$E(Z) = -Z\,\mathbf{C}\,Z + \sum_{i=1}^{m}\sum_{j=1}^{n}(1 - d_{ij}),$$

where $Z = \{Z_k; k = 1, ..., m+n\} = (X_1, ..., X_m, Y_1, ..., Y_n)$, i.e. ,

$$Z_k = \left\{ \begin{array}{lll} X_k & \text{for} & k \leq m \\ Y_{k-m} & \text{for} & m < k \leq m+n \end{array} \right.$$

and $\mathbf{C}$ is a symmetric matrix with the upper and diagonal part

$$\mathbf{C} = \left\{ \begin{array}{lll} C_{kk} = \tau_k^X & \text{for} & l = k \leq m \\ C_{kk} = \tau_k^Y & \text{for} & m < l = k \leq m+n \\ C_{kl} = \tau_{kl}^{XY} & \text{for} & k \leq m \quad \text{and} \quad m < l \leq m+n \\ C_{kl} = 0 & & \text{everywhere else} \end{array} \right. .$$

This formulation have lead us to believe this problem is NP hard, since in general, quadratic and non-positive form of integer problems are NP hard. However, because of the particular structure of the matrix $\mathbf{C}$, it is still possible that a polynomial algorithm exist.

2.1 Alternative views

Sometimes the representation of the strings (with cuts) is such that a cut must have occurred either at j or at $N-j+1$. More precisely, we can assume that the non-occurrence of a cut at j, $Y_j = 0$, implies that a cut occurred at $n-j+1$, i.e., $Y_{n-j+1} = 1$. We can then consider the following two costs

1. Maximize #(bits=1) in the cut column.

$$\begin{aligned} E(Y,X) &= -\sum_{i=1}^{m}\sum_{j=1}^{n/2} Y_j\left[(1-X_i)d_{ij} + X_i d_{i,n-j+1}\right] \\ &\quad + (1-Y_j)[(1-X_i)d_{i,n-j+1} + X_i d_{ij}] \end{aligned} \tag{2}$$

   This cost function becomes very similar to the previous one, except that the term τ_j^Y changes to $\sum_{i=1}^{m}(d_{i,n-j+1} - d_{i,j})$ and the constant is also changed.

2. Maximize #(bits=1) $-$ #(bits=0) in the cut column and #(bits=0) $-$ #(bits=1) in the no-cut column. The cost function is then defined as follows:

$$\begin{aligned} E(Y,X) &= -\sum_{i=1}^{m}\sum_{j=1}^{n/2} Y_j\left[(1-X_i)(2\,d_{ij}-1) + X_i(2\,d_{i,n-j+1}-1)\right] \\ &\quad + (1-Y_j)[(1-X_i)(1-2\,d_{ij}) + X_i(1-2\,d_{i,n-j+1})] \end{aligned} \tag{3}$$

   This cost function is again similar to the original one, except the coefficient $\tau_i^X = 0$.

All these problems are similar from the optimization point of view.

2.2 Modeling the error

We describe how to account for two sources of noise in the system, first due to false cuts and true misses that are introduced to the system and, second due to errors in the localization of the cuts.

False cuts and True misses: On many occasions molecules will be cut where no cut is expected and some cuts will not appear where they are expected. The incidence of these events can be tested prior to our modeling. Assume that p_T is the probability of a "true" cut not being present and p_F the probability of "false" cut appearing in the molecule. We can then reformulate the energy function as

$$\begin{aligned}
E(Y,X) &= -\sum_{i=1}^{m}\sum_{j=1}^{n/2}\Big\{\,[(1-Y_j)p_F + Y_j(1-p_T)]\,[(1-X_i)d_{ij} + X_i d_{i,n-j+1}] \\
&\quad +[Y_j p_T + (1-Y_j)(1-p_F)]\,[(1-X_i)(1-d_{ij}) + X_i(1-d_{i,n-j+1})]\Big\} \\
&= -\sum_{i=1}^{m}\sum_{j=1}^{n/2}[\,2Y_jX_i(1-p_T-p_F)(d_{i,n-j+1}-d_{i,j}) + X_i(1-2p_F)(d_{ij}-d_{i,n-j+1}) \\
&\quad +Y_j(1-p_T-p_F)(2d_{i,j}-1) + (1-d_{ij}) + p_F(2\,d_{i,j}-1)\,] \\
&= A' - \sum_{i=1}^{m}\sum_{j=1}^{n/2}\tau_{ij}^{XY'}Y_jX_i - \sum_{i=1}^{m}\tau_i^{X'}X_i - \sum_{j=1}^{n/2}\tau_j^{Y'}Y_j\,.
\end{aligned}$$

where $A' = (1-d_{ij}) + p_F(2\,d_{i,j}-1)$, and

$$\left\{\begin{array}{ccc}
\tau_i^{X'} & = & (1-2p_F)\sum_{j=1}^{n/2}(d_{i,j}-d_{i,n-j+1}) = (1-2p_F)\tau_i^X \\
\tau_j^{Y'} & = & (1-p_T-p_F)\sum_{i=1}^{m}(2d_{i,j}-1) = (1-p_T-p_F)\tau_j^Y \\
\tau_{ij}^{XY'} & = & 2(1-p_T-p_F)(d_{i,n-j+1}-d_{i,j}) = (1-p_T-p_F)\tau_{ij}^{XY}
\end{array}\right. .$$

Note that for the special case $p_F = p_T = 0$ we recover the error-free model, and for the other special case $p_T + p_F = 1$ the problem becomes completely ambiguous, i.e., any solution is equally good and we can no longer solve for Y_j.

Error in localization: Many times a cut may be not localized properly. The model is then modified so that an observation of a cut in one location supports a cut in a nearby location.

$$\begin{aligned}
E(Y,X) &= -\sum_{i=1}^{m}\sum_{j=1}^{n/2}\Big\{\,[(1-Y_j)p_F + Y_j(1-p_T)]\,\frac{1}{1\,+\,2\,\alpha} \\
&\quad [(1-X_i)(d_{ij} + \alpha(d_{i,j+1}+d_{i,j-1})) + X_i(d_{i,n-j+1} + \alpha(d_{i,n-j+2}+d_{i,n-j}))] \\
&\quad +[Y_jp_T + (1-Y_j)(1-p_F)]\,\frac{1}{1\,+\,2\,\alpha} \\
&\quad [(1-X_i)((1-d_{ij}) + \alpha((1-d_{i,j+1}) + (1-d_{i,j-1}))) \\
&\quad +X_i((1-d_{i,n-j+1}) + \alpha((1-d_{i,n-j+2}) + (1-d_{i,n-j})))]\Big\} \\
&= A'' - \sum_{i=1}^{m}\sum_{j=1}^{n/2}[\,\tau_{ij}^{XY''}Y_jX_i] - \sum_{i=1}^{m}\tau_i^{X''}X_i - \sum_{j=1}^{n/2}\tau_j^{Y''}Y_j\,,
\end{aligned}$$

where $\alpha < 1$ indicates the weight of nearby observed cuts, $A'' = [1 - \frac{1}{1+2\alpha}(d_{ij} + \alpha(d_{i,j+1} + d_{i,j-1}))] + p_F[2\frac{1}{1+2\alpha}(d_{ij} + \alpha(d_{i,j+1} + d_{i,j-1})) - 1]$, and

$$\left\{\begin{array}{lcc} \tau_i^{X''} & = & \frac{(1-2p_F)}{1+2\alpha}\sum_{j=1}^{n/2}(d_{ij} - d_{i,n-j+1}) + \alpha[(d_{i,j-1} - d_{i,n-j}) + (d_{i,j+1} - d_{i,n-j+2})] \\ \tau_j^{Y''} & = & \frac{(1-p_T-p_F)}{1+2\alpha}\sum_{i=1}^{m}(2d_{i,j} - 1) + \alpha((2d_{i,j+1} - 1) + (2d_{i,j-1} - 1)) \\ \tau_{ij}^{XY''} & = & \frac{2(1-p_T-p_F)}{1+2\alpha}[(d_{i,n-j+1} - d_{i,j}) + \alpha((d_{i,j-1} - d_{i,n-j}) + (d_{i,j+1} - d_{i,n-j+2}))] \end{array}\right. .$$

The important conclusion is that after these errors have been accounted for, the optimization problem remains the same: to find a set of $\{X_i, Y_j\}$ that optimizes a cost function of the form given by (1). Moreover, from the optimization standpoint, the constants A, A', A'' are irrelevant and can be neglected as we proceed.

3 Approximate Solution

Here we present our first approximation method, based on our previous optimization work on template fitting in a problem of junction detection in images [11]. Indeed, this approximation method was devised in conjuction with the formulation of the problem. Then we show a linear time algorithm that approximates a solution.

3.1 A dynamic programming algorithm

Let m be the number of molecules; P the number of pixels per intensity profile. We are seeking to solve C_{fP} $(\hat{C}_{fP})$ where f is the maximum number of confrags [5]

Let M^i_{jk} denote the cost of fitting a single confrag to points $j, \ldots, k$ of molecule i. Let C_{fn} denote the cost of fitting f confrags to the m molecules while cosidering the points $1, \ldots, n$.
If we ignore the String Flipping Problem, we get

$$C_{fn} = \left\{\begin{array}{ll} \sum_{i=1}^m \ M^i_{1n} & f = 1, \\ \min_{i<j}\left\{C_{f-1j} + \sum_{i=1}^m \ M^i_{j+1n}\right\} & \text{otherwise.} \end{array}\right.$$

In order to account for the flipping, we incorporate a heuristic in the the formulation to decide on the flip of each molecule. Let M^{iL}_{jk} denote the cost of fitting a single confrag to points $j \ldots k$ of molecule i, taking left as the direction; and similarly M^{iR}_{jk}. $\mathcal{C}^L$ and $\mathcal{C}^R$ are arrays to store the intermediate values as shown below.

[5] Confrags are a group of 0's in the string, bounded by 1's, representing a fragment of the DNA molecule that has not been cut

$$\hat{C}_{fn} = \begin{cases} \sum_{i=1}^{m} \min(M_{1n}^{iL}, M_{1n}^{iR}) & f = 1, \\ \quad \mathcal{C}_{fni}^{L} = M_{1n}^{iL} & \\ \quad \mathcal{C}_{fni}^{R} = M_{1n}^{iR} & \\ \min_{i<j} \left\{ \hat{C}_{f-1j} + \sum_{i=1}^{m} \left\{ \min(\mathcal{C}_{f-1ji}^{L} + M_{j+1n}^{iL}, \mathcal{C}_{f-1ji}^{R} + M_{j+1n}^{iR}) + \Delta_i^L + \Delta_i^R \right\} \right\} & \text{otherwise.} \\ \quad \mathcal{C}_{fni}^{L} = \mathcal{C}_{f-1ji}^{L} + M_{j+1n}^{iL} + \Delta_i^L & \\ \quad \mathcal{C}_{fni}^{R} = \mathcal{C}_{f-1ji}^{R} + M_{j+1n}^{iR} + \Delta_i^R & \end{cases}$$

where

$$\Delta_i^L = \begin{cases} \text{Extra (non-negative) cost} & \text{if molecule } i \text{ flipped,} \\ 0 & \text{otherwise.} \end{cases}$$

Δ_i^R is defined similarly.

The formulation is based on the profile, but it is both feasible and practical to preprocess the input so as to locate positions of cuts with some probability on the profile and work on the data as being a sequence of real numbers denoting cut or no cut with some confidence.

3.2 A linear time algorithm

This is a linear time approximate algorithm (that does $O\left(n(m + \log n)\right)$ work) which has an approximation factor > 0.5. This algorithm works from left to right (or reverse) with respect to the data strings. It makes assumptions about the structure of the problem that are not valid and consequently is biased towards the orientation of the solution. However, some preliminary experiments suggest that it may be a useful algorithm.

Algorithm: We have used the energy equation (2).

Let us define $j' = n - j + 1$ for simplicity.

1. Let $c_j = \sum_i^m \text{cnt}(d_{ij}, d_{ij'})$, $j = 1, 2, \ldots, n/2$, where, $\text{cnt}(x, y) = \begin{cases} 0 & x = y = 0, \\ 1 & \text{otherwise.} \end{cases}$

   $O(mn)$ work.

2. Sort $c_j, j = 1, 2, \ldots, n/2$ to obtain s_j.

   $O(n \log n)$ work.

3. Define loss$[i]$ = gain$[i]$ = 0, parity$[i]$ = no-flip, $i = 1, 2, \ldots, m$. Define G and G' as follows. Initialize $G = G' = 0$.

   For $j = 1, 2, \ldots, n/2$, do the following:

   3-1. Assume the cut position of j and, count the increase in support of this hypothesis.

   if parity$[i]$ = flip,

```
                if $d_{ij'} = 1$ then tgain = gain[$i$] + 1 else tloss = loss[$i$] + 1.
        if parity[$i$] = no-flip,
                if $d_{ij} = 1$ then tgain = gain[$i$] + 1 else tloss = loss[$i$] + 1.
        If (tloss > tgain)
              then $G = G +$ (tloss − gain[$i$])
              else $G = G +$ (tgain − gain[$i$]).

    3-2. Assume the cut position of $j'$, and, count the increase in support of this hypothesis.
        if parity[$i$] = no-flip,
                if $d_{ij'} = 1$ then tgain = gain[$i$] + 1 else tloss = loss[$i$] + 1.
        if parity[$i$] = flip,
                if $d_{ij} = 1$ then tgain = gain[$i$] + 1 else tloss = loss[$i$] + 1.
        If (tloss > tgain)
              then $G' = G' +$ (tloss − gain[$i$])
              else $G' = G' +$ (tgain − gain[$i$]).

    3-3. If $G > G'$, position $j$ is a cut else $j'$. Update loss[$i$], gain[$i$], parity[$i$], $i = 1, 2, \ldots, m$ accordingly.
```

$O(m)$ work.

Approximation Factor: Let c_j be as defined in the algorithm. Then,

$$c_1 + c_2 + \ldots + c_{n/2} \geq E^{\text{optimal}}.$$

Clearly, from the algorithm,

$$E^{\text{approx}} \geq c_1 + \frac{c_2 + c_3 + \ldots + c_{N/2}}{2}.$$

Thus, the approximation factor > 0.5.

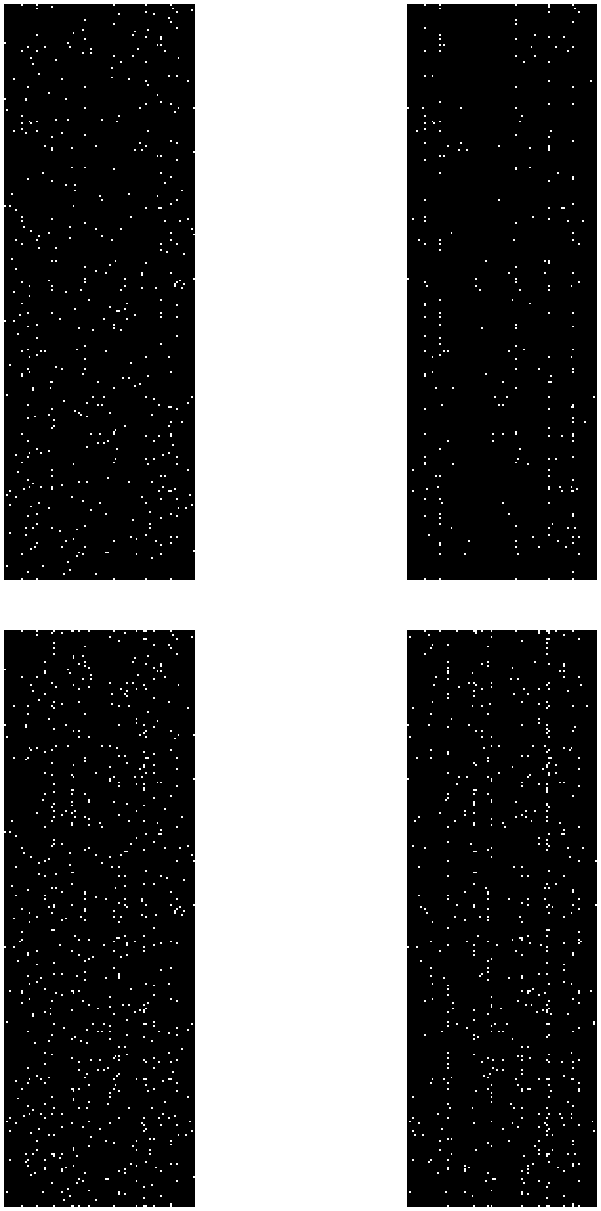

Figure 1: Examples using the linear algorithm: The images on the left are the input to the flipping string problem with white marks representing the cuts. The images on the right are the solution with the best flip for each molecule. [m=300, n=100]

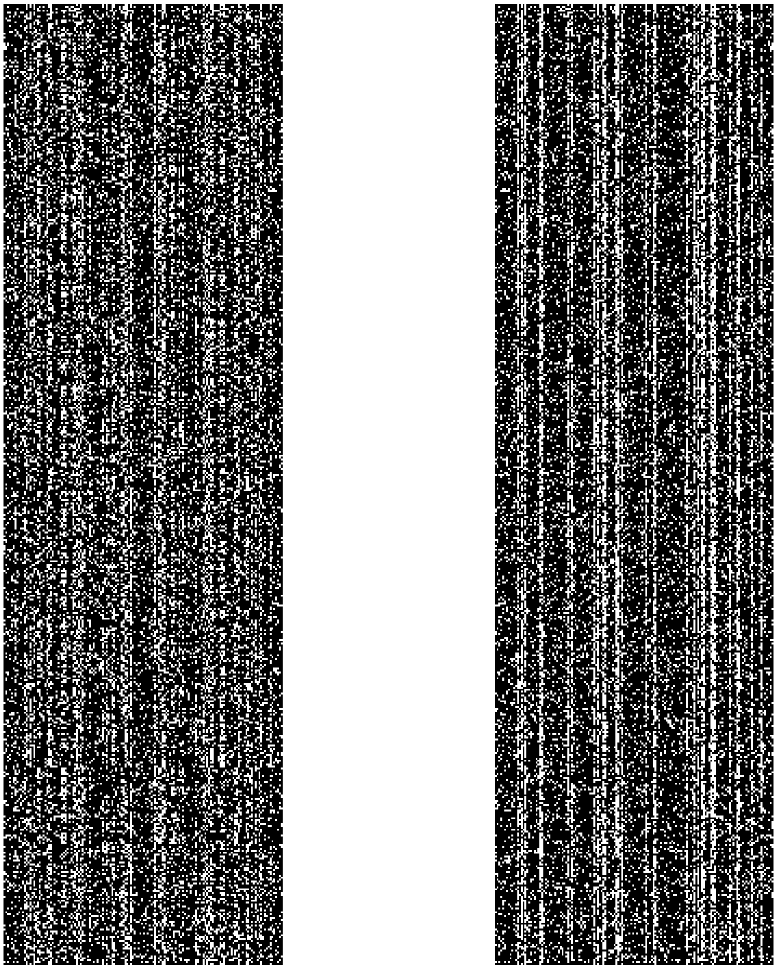

Figure 2: Another example, more realistic one to include data errors, of using the linear algorithm: Rate of false negative (p_T) is 0.3, false positive (p_F) is 0.16, number of molecules (m) 500, length of each molecule (n) is 150 pixels. The algorithm extracts the hypothesis that was used to create the data. The column on the right representes the flipped strings producing a "maximally" correlated data.

4 Statistical approach: MPM and MAP

It is known that from the estimation of Z we can obtain average estimates of X and Y. These average estimates correspond to the MPM estimate of the posterior distribution. Moreover, it is known (e.g, [4]) that in the limit of $\beta \to \infty$ the MPM estimate becomes the MAP estimate. So we address the two problems, the one of estimating the MPM estimates, $\bar{X}_i$ and $\bar{Y}_j$, and the MAP estimates. More precisely, we have

$$\begin{aligned} \bar{X}_i &= \frac{1}{\beta}\sum_{X,Y} X_i P(X,Y) = \frac{1}{\beta}\frac{\partial logZ}{\partial \tau_i^X} \\ \bar{Y}_j &= \frac{1}{\beta}\sum_{X,Y} Y_j P(X,Y) = \frac{1}{\beta}\frac{\partial logZ}{\partial \tau_i^Y}\,, \end{aligned} \tag{4}$$

where $\bar{x}$ is the average estimate of x. We now proceed to compute Z.

$$\begin{aligned} Z &= \sum_{\{X_i=0,1\}^m} \sum_{\{Y_j=0,1\}^{n/2}} \prod_{i=1}^{m}\prod_{j=1}^{n/2} e^{\beta[\frac{\tau_i^X}{n/2}X_i + \frac{\tau_j^Y}{m}Y_j + \tau_{ij}^{XY}X_iY_j]} \\ &= \sum_{\{Y_j=0,1\}^{n/2}} \prod_{i=1}^{m} \sum_{X_i=0,1} \prod_{j=1}^{n/2} e^{\beta[\frac{\tau_i^X}{n/2}X_i + \frac{\tau_j^Y}{m}Y_j + \tau_{ij}^{XY}X_iY_j]} \\ &= \sum_{\{Y_j=0,1\}^{n/2}} \prod_{i=1}^{m}(\prod_{j=1}^{n/2} e^{\beta\frac{\tau_j^Y}{m}Y_j} + \prod_{j=1}^{n/2} e^{\beta(\frac{\tau_i^X}{n/2} + \frac{\tau_j^Y}{m}Y_j + \tau_{ij}^{XY}Y_j)}) \\ &= \sum_{\{Y_j=0,1\}^{n/2}} \prod_{i=1}^{m}(\prod_{j=1}^{n/2} e^{\beta\frac{\tau_j^Y}{m}Y_j})(1 + \prod_{j=1}^{n/2} e^{\beta(\frac{\tau_i^X}{n/2} + \tau_{ij}^{XY}Y_j)}) \end{aligned}$$

analogously we can average out $\{Y_j\}$

$$= \sum_{\{X_i=0,1\}^m} \prod_{j=1}^{n/2}(\prod_{i=1}^{m} e^{\beta\frac{\tau_i^X}{n/2}X_i})(1 + \prod_{i=1}^{m} e^{\beta(\tau_j^Y + \tau_{ij}^{XY}X_i)})\,. \tag{5}$$

In appendix A we investigate on the factorization property of multivariable functions, possibly related to the NP hardness of this problem.

4.1 Mean field approximation and annealing

We can now apply the mean field approximation, by fixing some of the variables at the unknown mean state, and averaging out the remaining ones as follows

$$
\begin{aligned}
Z &= \sum_{\{Y_j=0,1\}^{n/2}} (\prod_{j=1}^{n/2} e^{\beta \frac{\tau_j^Y}{m} Y_j})^m \prod_{i=1}^{m} (1 + \prod_{j=1}^{n/2} e^{\beta(\frac{\tau_i^X}{n/2} + \tau_{ij}^{XY} Y_j)}) \\
&\approx (\prod_{j=1}^{n/2} \sum_{Y_j=0,1} e^{\beta \tau_j^Y Y_j}) \prod_{i=1}^{m} (1 + \prod_{j=1}^{n/2} e^{\beta(\frac{\tau_i^X}{n/2} + \tau_{ij}^{XY} \bar{Y}_j)}) \\
&\approx (\prod_{j=1}^{n/2} (1 + e^{\beta \tau_j^Y})) \prod_{i=1}^{m} [1 + e^{\beta \tau_i^X} \prod_{j=1}^{n/2} e^{\beta \tau_{ij}^{XY} \bar{Y}_j}] \, , \qquad (6)
\end{aligned}
$$

where $\bar{Y}_j$ is the mean value of Y_j, that is still unknown. Analogously, we obtain

$$
Z \approx (\prod_{i=1}^{m} (1 + e^{\beta \tau_i^X})) \prod_{j=1}^{n/2} [1 + e^{\beta \tau_j^Y} \prod_{i=1}^{m} e^{\beta \tau_{ij}^{XY} \bar{X}_i}] \, , \qquad (7)
$$

where $\bar{X}_i$ is the mean value of X_i that it is still unknown. We can now obtain the average estimate of the variable X_i from equations (4) and (6) as follows

$$
\begin{aligned}
\bar{X}_i &= \frac{e^{\beta \tau_i^X} \prod_{j=1}^{n/2} e^{\beta \tau_{ij}^{XY} \bar{Y}_j}}{[1 + e^{\beta \tau_i^X} \prod_{j=1}^{n/2} e^{\beta \tau_{ij}^{XY} \bar{Y}_j}]} = \frac{e^{\beta \sum_{j=1}^{n/2} \tau_{ij}^{XY} \bar{Y}_j}}{[e^{-\beta \tau_i^X} + e^{\beta \sum_{j=1}^{n/2} \tau_{ij}^{XY} \bar{Y}_j}]} \\
&= \frac{1}{(e^{\beta \sum_{j=1}^{n/2} \tau_{ij}^{XY} (\frac{1}{2} - \bar{Y}_j)} + 1)} \, , \qquad (8)
\end{aligned}
$$

where we used $\tau_i^X = -\frac{1}{2} \sum_{j=1}^{n/2} \tau_{ij}^{XY}$. Note that the limit $\beta \rightarrow \infty$ we have

$$
\lim_{\beta \rightarrow \infty} \bar{X}_i = \Theta(\sum_{j=1}^{n/2} \tau_{ij}^{XY} (\bar{Y}_j - \frac{1}{2})) \, , \qquad (9)
$$

where $\Theta(x) = 1$ if $x \geq 0$ and $\Theta(x) = 0$ otherwise. This is equivalent to optimizing the energy (1) for the X_i's variables while keeping the Y_j's fixed.

Analogously we obtain for the average estimate of Y_j, from equations (4) and (7),

$$
\bar{Y}_j = \frac{e^{\beta \sum_{i=1}^{m} \tau_{ij}^{XY} \bar{X}_i}}{[e^{-\beta \tau_j^Y} + e^{\beta \sum_{i=1}^{m} \tau_{ij}^{XY} \bar{X}_i}]} \, . \qquad (10)
$$

In the limit $\beta \rightarrow \infty$ we have

$$
\lim_{\beta \rightarrow \infty} \bar{Y}_j = \Theta(\tau_j^Y + \sum_{i=1}^{m} \tau_{ij}^{XY} \bar{X}_i) \, . \qquad (11)
$$

This is equivalent to optimize the energy (1) for the Y_j's variable while keeping the $X_i^{\prime}s$ fixed.

Algorithm: We can now devise an approach to find an approximate solution to the problem. We start with $\beta = 0$ where the solution is clearly $\bar{X}_i = \bar{Y}_j = 1/2$ for all i and j.

We then increment β and evaluate the new set of $\{\bar{X}_i\}$ and $\{\bar{Y}_j\}$ by iterations over (8) and (10), until a stable point is reached. We then proceed and increment β and update $\{\bar{X}_i\}$ and $\{\bar{Y}_j\}$ again. In the limit of $\beta \rightarrow \infty$ the solution $\{\bar{X}_i\}$ and $\{\bar{Y}_j\}$ becomes $0, 1$ and a solution is obtained.

Acknowledgement

This work is motivated by the activities on the DNA optical mapping developed at the W.M.Keck Laboratory for Biomolecular Imaging, led by David Schwartz.

A A remark on function factorization

Let us consider the partition function again

$$Z = \sum_{\{Y_j=0,1\}^{n/2}} \prod_{i=1}^{m} (\prod_{j=1}^{n/2} e^{\beta \frac{\tau_j^Y}{m} Y_j})(1 + \prod_{j=1}^{n/2} e^{\beta(\frac{\tau_i^X}{n/2} + \tau_{ij}^{XY} Y_j)}) .$$

Suppose that we could factorize

$$(1 + \prod_{j=1}^{n/2} e^{\beta(\frac{\tau_i^X}{n/2} + \tau_{ij}^{XY} Y_j)}) = \prod_{j=1}^{n/2} f_{ij}(Y_j) .$$

We also assume that such a factorization process would not take exponential time in n. Then we would obtain

$$\begin{aligned} Z &= \sum_{\{Y_j=0,1\}^{n/2}} \prod_{i=1}^{m} \prod_{j=1}^{n/2} e^{\beta \frac{\tau_j^Y}{m} Y_j} f_{ij}(Y_j) \\ &= \prod_{j=1}^{n/2} \sum_{Y_j=0,1} \prod_{i=1}^{m} e^{\beta \frac{\tau_j^Y}{m} Y_j} f_{ij}(Y_j) \\ &= \prod_{j=1}^{n/2} (\prod_{i=1}^{m} f_{ij}(0) + \prod_{i=1}^{m} f_{ij}(1) e^{\beta \frac{\tau_j^Y}{m}}) . \end{aligned}$$

This would be an exact/analytical calculation for Z and so $\bar{X}_i$ and $\bar{Y}_j$ could be directly computed from (4). The complexity would be the one to compute the coefficients τ_i^X, τ_j^Y and tau_{ij}^{XY}, which is linear on n and on m, multiplied by the complexity of the calculation of Z which is also linear on n and m (assuming that the factorization process is constant). This would produce a $O(n^2m^2)$ algorithm.

However we can show that such decomposition, $(1+\prod_{j=1}^{n/2} e^{\beta(\frac{\tau_i^X}{n/2}+\tau_{ij}^{XY}Y_j)}) = \prod_{j=1}^{n/2} f_{ij}(Y_j)$, does not occur.[6].

Proof: Suppose it does occur, and that the functions are positive, then

$$\sum_{j=1}^{n/2} \log(f_{ij}(Y_j)) = \log(1+\prod_{j=1}^{n/2} e^{\beta(\frac{\tau_i^X}{n/2}+\tau_{ij}^{XY}Y_j)})$$

and the Hessian $(\frac{\partial^2}{\partial Y_k \partial Y_l})$ *on both sides becomes*

$$\begin{aligned}
\delta_{kl}\frac{\partial^2}{\partial Y_k^2}\log(f_{ik}(Y_k)) &= \frac{\partial^2}{\partial Y_k \partial Y_l}\log(1+\prod_{j=1}^{n/2} e^{\beta(\frac{\tau_i^X}{n/2}+\tau_{ij}^{XY}Y_j)}) \\
&= \beta\frac{\partial}{\partial Y_l}\frac{\tau_{ik}^{XY}\prod_{j=1}^{n/2} e^{\beta(\frac{\tau_i^X}{n/2}+\tau_{ij}^{XY}Y_j)}}{(1+\prod_{j=1}^{n/2} e^{\beta(\frac{\tau_i^X}{n/2}+\tau_{ij}^{XY}Y_j)})} \\
&= \beta^2\tau_{ik}^{XY}\tau_{il}^{XY}[\prod_{j=1}^{n/2} e^{\beta(\frac{\tau_i^X}{n/2}+\tau_{ij}^{XY}Y_j)}(1+\prod_{j=1}^{n/2} e^{\beta(\frac{\tau_i^X}{n/2}+\tau_{ij}^{XY}Y_j)}) \\
&\quad -(\prod_{j=1}^{n/2} e^{\beta(\frac{\tau_i^X}{n/2}+\tau_{ij}^{XY}Y_j)})^2]\,/\,(1+\prod_{j=1}^{n/2} e^{\beta(\frac{\tau_i^X}{n/2}+\tau_{ij}^{XY}Y_j)})^2 \\
&= \beta^2\tau_{ik}^{XY}\tau_{il}^{XY}\prod_{j=1}^{n/2} e^{\beta(\frac{\tau_i^X}{n/2}+\tau_{ij}^{XY}Y_j)}/\,(1+\prod_{j=1}^{n/2} e^{\beta(\frac{\tau_i^X}{n/2}+\tau_{ij}^{XY}Y_j)})^2 \\
&= \beta^2\tau_{ik}^{XY}\tau_{il}^{XY}F(\{Y_j\})
\end{aligned}$$

And this is a contradiction, since the left hand side must be zero for $k \neq l$ and the right hand side is not zero.

Now we offer a different proof, without using the assumption of a positive decomposition,

Proof: Let us first consider the problem of two variables, i.e., $1+f(x)g(y) = r(x)s(y)$.

Then for a discrete set of values of x and y (indeed we are only interested in the cases $x, y \in \{0,1\}$) we have $1+f_i\,g_j = r_i s_j$, where $i = 0,1...,I$ and $j = 0,1,...,J$. The matrix $M_{ij} = r_i s_j$ has rank 1 since every column jth of M_{ij} is a scalar (s_j) times a vector $s = s_0, s_1, ..., s_I$. So all columns are linear dependent. The condition for the matrix $A_{ij} = 1+f_i\,g_j$ to have rank 1 is that the columns and rows become linear dependent. This occurs only if $g_j = 0\,\forall j$ or if $f_i = 0\,\forall i$ or if f_i and g_j are constant. Thus, the decomposition is not valid.

[6] One of the authors have profited from conversations with L. Gurvits on this proof

In the case of more variables, say three, we have $1 + f(x)g(y)h(z) = r(x)s(y)t(z)$*. Then we can group two variables to have* $1 + f(x)G(y, z) = r(x)S(y, z)$*. Then, we can sample again and consider a matrix where the columns represent all samples on* y *and* z*. The matrix is not square, but the results are the same as the ones we obtained above. Therefore, the impossibility of the factorization is generalizable to any number of variables.*

Therefore, the sum of these two functions is not factorizable. More importantly, it suggest that (if indeed the original problem is NP hard) the hardness of NP problems may be related to the impossibility of this factorization.

References

[1] Cooper N. G. (editor), *The Human Genome Project - Deciphering the Blueprint of Heredity*, University Science Books, Mill Valley, California, 1994.

[2] Wang Y., Huff E., Schwartz D., *Optical Mapping of site-directed cleavages on single DNA molecules by the RecA-assisted restriction endonuclease technique*, Proc. Nat. Acad. Sci., 92, pp 165-169, January 1995.

[3] Gnedenko B. V., *The Theory of Probability*, Chelsea Publishers, New York, 1962, pp 122-128.

[4] D. Geiger and F. Girosi, *Parallel and deterministic algorithms for MRFs: surface reconstruction*, Pattern Analysis and Machine Intelligence, 5, pp 401-412, May 1991.

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

[6] Idury R. M., Waterman M. S., *A New Algorithm for DNA Sequence Assembly*, J. of Comp. Bio. 2(2), pp 291-306, 1995.

[7] Lander E. S., Waterman M. S., *Genomic mapping by fingerprinting random clones: a mathematical analysis*, Genomics, 2, pp 231-239, 1988.

[8] J.-S.Lo. *Detection and Analysis of DNA from Microscope Images.* Master Thesis at the Department of Computer Science, Courant Institute, NYU. May 1995.

[9] Meng X., Benson K., Chada K., Huff E., Schwartz D., *Optical mapping of lambda bacteriophage clones using restriction endonucleases*, Nature Genetics, 9, pp 432-438, April 1995.

[10] D. Schwartz, X. Li, L. Hernandez, S. Ramnarain, E. Huff, Y. Wang. *Ordered Restriction Mapx of Saccharomyces cerevisiae Chromosomes Constructed by Optical Mapping*, In Proc. Natl. Acad. Sci. USA, 92:165-169, 1995.

[11] Laxmi Parida, Davi Geiger, Robert Hummel, *Junction Detection Using Piecewise Constant Functions*, (submitted for publication).