On the K-Populations Problem *

Laxmi Parida

Computational Biology Center
IBM Thomas J. Watson Research Center
Yorktown Heights, NY10598, USA

Bud Mishra†

Department of Computer Science
Courant Institute, New York University
New York, NY10012, USA

Abstract

Given a set of m molecules, derived from K homologous clones, we wish to partition these molecules into K populations, each giving rise to distinct ordered restriction maps, thus providing simple means for studying biological variations. With the emergence of single molecule methods, such as optical mapping, that can create individual ordered restriction maps reliably and with high throughput, it becomes interesting to study the related algorithmic problems. In particular, we provide a complete *computational complexity analysis of the "K-populations" problem along with a probabilisitc analysis. We also present some simple polynomial heuristics, while exposing the relations among various error sources that the optical mapping approach may need to cope with. We believe that these results will be of interest to computational biologists in devising better algorithms, to biochemists in understanding the trade-offs among the error sources and finally, to biologists in creating reliable protocols for population study.*

1 Introduction

Optical mapping [AMS97, Cai+95, GP96, Men+95, MP96, Sam+95, Sch+93, WHS95] provides a practical high-throughput approach to restriction-based genomic analysis. The ordered restriction maps from optical mapping provide actual physical distances between the markers and possess unambiguous structural information, whose accuracy and resolution can be actively controlled by increasing the number of restriction enzymes and the number of individual molecules. Optical mapping finds applications in contiging (aligning overlapping sets of cloned DNA or genomic DNA), identifying genetic loci, sequence anchoring and verification, and in population genetic studies. Here, we explore some algorithmic and complexity questions related to the use of optical mapping to study a population of homologous

*An extended abstract of this paper appeared in the Proceedings of the Second Annual Conference on Computational Molecular Biology (RECOMB98) [PM98].

† *The research presented here was partly supported by an NIH Grant: NIH R01 HG0025-07.*

DNA fragments. In practice, the most interesting case involves just two populations ($K = 2$) where one examines populations related to diploid DNA. However, there are other situations, e.g., dealing with PCR products, several strains of a microorganisms, etc., where K could be arbitrarily large. Computationally, it is also interesting to study how this generality (when K is unconstrained) affects the problem.

We focus on a somewhat idealized model. We assume that we are given m molecules, each one derived from one of K different clones. We wish to partition the m molecules into K different disjoint classes (K "populations"), each class corresponding to a distinct clone. We then wish to compute and output the K distinct ordered restriction maps (one for each population) so that the dissimilarities among the ordered restriction maps can be quickly distinguished. The output is given in a novel data structure that can quickly identify the dissimilar regions in the maps.

In an ideal setting, if the correct ordered restriction map for each molecule can be made available, then the resulting computational problem is rather trivial. In practice, however, the single molecule restriction map that can be computed by the image processing algorithm will be governed by several error processes: missing restriction cut sites due to partial digestion (false negatives), spurious optical cut sites (false positive), sizing error, missing fragments, error in assigning the orientation of the molecule, presence of other spurious molecules etc.

Even though we ignore all but the first two error processes, the resulting computational problem still poses many challenges from a purely algorithmic point of view. We characterize these structural difficulties, propose an extension of an earlier heuristic and explore its power empirically.

For the purpose of complexity analysis, we further simplify our model to expose the underlying combinatorial structure. This simplification results in making the arguments showing the SNP-hardness of the problem easier to follow. It is straightforward to note that the more realistic model only makes the computational complexity worse. In our simplified model, each molecule (m in total) is represented as a binary vector of length n:

$$A_i = (a_{i1}, a_{i2}, \ldots, a_{in}) \in \{0,1\}^n,$$

and the set of m molecules is represented by an $m \times n$ 0-1 matrix. A 1 in location a_{ij} is meant to indicate a cut site at the location j in the i^{th} molecule and may be a true restriction cut or a false optical cut. Thus, if $a_{ij} = 1$, then either it is a true restriction site and the correct map for the corresponding clone has a restriction site at j or it is a false optical cut and the correct map has no restriction site at j. Conversely, if $a_{i,j} = 0$, then either it is a missing restriction site and the correct map has a restriction site at j or it is simply not a restriction site and the correct map does not have a restriction site at j. Our goal is to devise an algorithm to find a partition of the molecules into K populations so that each restriction site in the proposed map for a population is supported by "enough" restriction sites at the corresponding location in the same population. We shall define a cost function that formalizes this notion and examine the complexity of the resulting combinatorial optimization problem.

The paper is organized as follows: In the next section, we reformulate the problem in a purely combinatorial setting and show that the resulting problem as well as computing an arbitrarily good approximation is computationally infeasible (NP-hard). In Section 3, we give a 0.756-approximation algorithm for a 2-populations problem. In Section 4 we give a probabilistic analysis of the problem and in Section 5, we propose a simple heuristic assuming an oracle that can create the complete all-population map and demonstrate experimentally that the resulting simple polynomial time algorithm finds the maps correctly for reasonable values of the parameters (partial digestion rate of 50%, negligibly small optical false cut rate, upto 6 populations). In Section 6, we describe our empirical results based on the synthesized data and explore its limitations. In a concluding section, we interpret the significance of our results.

2 Complexity

For the sake of complexity analysis, we may assume that the only errors to be handled are false negative and false positive errors (due to partial digestion and optical cuts, respectively[1]). Given a set of molecules from K different populations, the task is to identify the different populations and the map of each of the population. The problem can be formally stated as follows. We are given a $m \times n$ matrix $[a_{ij}]$ denoting m molecules and n sites with p_j, $j = 1, 2, \ldots, n$ defined for each site j, which is a lower bound on the fraction of the size of the population where the site j is a consensus cut[2]. The task is to maximize the number of 1's in the consensus cut columns and the number of 0's in the non-consensus cut columns in each population.

Let Y_{11}, Y_{12}, $\ldots$, Y_{1n}, Y_{21}, Y_{22}, $\ldots$, Y_{2n}, $\ldots$, Y_{K1}, Y_{K2}, $\ldots$, Y_{Kn} be the (nK) indicator variables associated with each site (column) and population with the following connotation:

$$Y_{kj} = \begin{cases} 1, & \text{if } j \text{ is a consensus cut site in pop } k; \\ 0, & \text{if } j \text{ is not a consensus cut site in pop } k. \end{cases}$$

$\mathrm{Pop}\,(A_i) = k$ where $1 \le k \le K$, denotes that molecule i belongs to population k. Let

$$X_{ik} = \begin{cases} 1, & \text{if } \mathrm{Pop}\,(A_i) = k; \\ 0, & \text{otherwise.} \end{cases}$$

[1]Recall that we may also model sizing errors or uncertainty in orientation or errors due to missing fragments or spurious molecules, etc. However, these errors only result in higher worst-case complexity of the problem and complicate the constructions used in the proof.

[2]In general, p_j depends on the digestion rate, which cannot be always expected to be known a priori, and may have to be estimated from the data (for instance, by an MLE method). A simple heuristic estimator can be formulated by assuming that all the p_j's are equal and is given by

$$p = (1-\lambda)(\tilde{p}/m)\frac{\sum \#\text{cuts in} A_i}{\text{length}(A_i)},$$

where $\tilde{p}$ denotes the cutting efficiency (e.g., $\approx 1/4,000$ for a 6-cutter) and $(1-\lambda)$ is a shrinkage factor that compensates for the bias due to false optical cuts.

and $m_k = \sum_{i=1}^{m} X_{ik}$. The K-populations problem then can be formulated as a combinatorial optimization problem; *maximize the following function*:

$$C(p_j, [a_{ij}], X_{ik}, Y_{kj}) = \sum_{k=1}^{K} \left\{ \sum_{j=1}^{n} \sum_{i=1}^{m} Y_{kj} [X_{ik} a_{ij} - p_j m_k] \right\}. \tag{1}$$

and the *cost* of the configuration (given by optimizing the above function) is the total number of 1's in the consensus cut columns of each population. Note that the function given by equation (1) is simply

$$\sum_{k=1}^{K} \sum_{j=1}^{n} \sum_{i=1}^{m} X_{ik} Y_{kj} [a_{ij} - p_j].$$

The cost function used here is a straightforward generalization of the "consensus with data" cost function used for the 1-population problem described in [Par97b]. This is somewhat simplified by the fact that we include only those terms that model the false positive and false negative errors. This cost function can also be heuristically justified by considering its expected value over random $m \times n$ 0-1 matrices, where the entries of the matrix are Bernoulli random variables with negligibly small false positive probability and with a false negative probability determined by p_j's. The proofs of the following theorems are along the lines of the SNP-hardness proofs of the Binary Flip Cut (BFC) problem and its variants in [Par97a]. However the proof here is self-contained and incorporates modifications required for the K-populations problem.

Theorem 1 *The K-populations problem is NP-hard. Further, there exists a constant $\epsilon > 0$ such that approximating the problem within a factor of $1 - \epsilon$ is NP-hard.*

Proof Sketch. We will prove the result for a special case of the K-populations problem with the following characteristics:

1. $K = 2$. (A two-populations problem.)
2. $\sum_{k=1}^{K} Y_{kj} = 1$ for all j. (Every cut belongs to exactly one population.)
3. The two populations are of the same size. Notice that without a constraint on the sizes of the populations, we can obtain trivial configurations by assigning zero size to one population. We further assume that we are also given the molecules in the following order: the molecules come in pairs such that no two elements of the pair belong to the same population.
4. $p_j = \sum_i a_{ij}/2m$ for all j, where m is the total number of molecules.

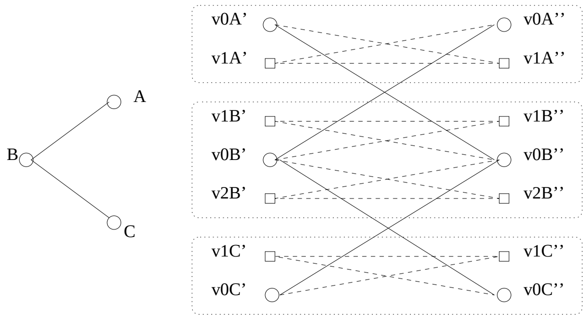


Figure 1: The graph for the MC problem is shown on the left. The corresponding graph for the the BMC problem is shown on the right where an edge with positive weight is shown as a solid line and the one with a negative weight is shown as a dashed line. The gadgets corresponding to each of the vertices in the graph for the MC problem are enclosed in dotted boxes.

To prove the inapproximability of this special case of the K-populations (KP) problem [3], we use the recent technique of giving a *gap-preserving reduction* of a Max SNP-hard problem, the Max-cut (MC) problem, to our problem [AL97].

The proof has three steps. In step 1 we show the reduction of an instance of the MC problem to an instance of a bipartite Max-cut problem, with weights on the edges as 1 or -1 (BMC). In step 2 we show the reduction of an instance of the BMC problem to an instance of the KP problem and in the final step we show that the reduction is *gap-preserving*.

Let C^*_X denote the cost of the optimal solution and $\tilde{C}_X$ denote the cost of an approximate solution of the problem X.

Step 1. (MC to BMC reduction.)
Consider an MC problem on a graph with vertices and edges $(V, E), n = |V|, e = |E|$. Construct an instance of BMC with $(\tilde{V}, \tilde{E})$ as follows: For each $v_i \in V$, with degree d_i, construct $2(d_i + 1)$ vertices, $V_{gadget_i} = \{v'_{i0}, v'_{i1}, \ldots, v'_{id_i}, v''_{i0}, v''_{i1}, \ldots, v''_{id_i}\}$. Further, $wt(v'_{ij}, v''_{ij}) = wt(v'_{i0}, v''_{ij}) = wt(v'_{ij}, v''_{i0}) = -1, j = 1, 2, \ldots, d_i$. Thus, v_i gives rise to $3d_i$ edges with negative weight. Also if $v_1 v_2 \in E$ then $wt(v'_{10} v''_{20}) = wt(v'_{20} v''_{10}) = +1$. It can be seen that this construction gives a bipartite graph with $\tilde{V} = V' \cup V''$ where $v'_x \in V', v''_x \in V''$. See Figure 1 for an example.

Thus the BMC has $2n + 6e$ vertices, and, $2e$ edges with weights $+1$, and, $6e$ edges with negative weights. Recall for any graph $\sum_i d_i = 2e$.

Let a solution be of size K, and, the partition of the vertices induced by this solution be S_1 and S_2 in the MC problem. We make the following observations.

1.1 In a solution of the BMC, the two sets $\tilde{S}_1, \tilde{S}_2$, are such that $V_{gadget_i} \subset \tilde{S}_1$ or $\tilde{S}_2$, $\forall i$. If this does not hold, that is $V_{gadget_i} \not\subset \tilde{S}_1$, then the solution can be modified that only improves the solution. In a solution to the BMC, if $v'_1 v''_2$ is in the cut, so must $v''_1 v'_2$

[3]The argument given here is similar to the Max SNP-hardness proof of the Exclusive Binary Flip-Cut (EBFC) problem discussed in [Par97a].

(called the *image* of $v_1'v_2''$). This follows as V_{gadget_1} and V_{gadget_2} are in the sets $\tilde{S}_1$ and $\tilde{S}_2$ respectively (without loss of generality). Hence, given a solution to the BMC, the solution to the corresponding MC is constructed as follows: if $v_1'v_2''$ (and its image) is in the solution to the BMC, then v_1v_2 is in the solution to the MC.

1.2 We make the following claim:

$$\begin{aligned} C^*_{MC} &= C^*_{BMC}/2 \\ \tilde{C}_{MC} &\geq \tilde{C}_{BMC}/2 \end{aligned} \tag{2}$$

This is easy to see from the construction of the instance of the BMC problem.

Step 2. (BMC to KP reduction.)
Consider a BMC $((V_1, V_2), E)$, $V_1 = \{v_1^1, v_2^1, \ldots, v_m^1\}$, $V_2 = \{v_1^2, v_2^2, \ldots, v_n^2\}$. Define $\bar{i} = m + i$. Construct an instance of KP $[M_{ij}]$ with $2m$ rows and n columns as follows. If $wt(v_i^1 v_j^2) = 1$, then $M_{ij} = 1, M_{\bar{i}j} = 0$. If $wt(v_i^1 v_j^2) = -1$, then $M_{ij} = 0, M_{\bar{i}j} = 1$. If $v_i^1 v_j^2$ is not an edge in the BMC, then $M_{ij} = M_{\bar{i}j} = 0$. See Figure 2 for an example. This construction ensures that molecules/rows corresponding to i and $\bar{i}$ belong to different populations (this also explains the third characteristic of the special KP problem). Also notice that, assuming that the assignment of the rows/molecules to the populations have been made, a consensus cut column is the one with the larger number of 1's (this explains the fourth characteristic of the special KP problem).

We make the following observations.

2.1 Given a a configuration (assignments of populations one/two to rows and cuts/no-cuts to columns) for the KP problem, it can be shown that we can obtain a solution for the BMC problem. We can further show that given an approximate solution for the KP problem, we can construct an approximate solution for the BMC problem and any solution is $\geq 6e$. Also the solution to the KP problem is such that the molecules/rows and sites/columns corresponding to the vertices of a gadget of a vertex from the MC belong to the same population. Thus the corresponding solution for the BMC (and then the MC) can be obtained.

2.2 We make the following claim:

$$\begin{aligned} C^*_{BMC} &= C^*_{KP} - 6e \\ \tilde{C}_{BMC} &\geq \tilde{C}_{KP} - 6e \end{aligned} \tag{3}$$

This holds since the number of edges with negative weights is $6e$ in the BMC instance. (The reader may see [Par98, Par97a] for the details of the argument.)

Step 3. (*Gap-preserving* reduction.)
Finally, we show that the reduction is *gap-preserving*.

	v0A"	v1A"	v0B"	v1B"	v2B"	v0C"	v1C"	
v0A'	0	0	1	0	0	0	0	2
v1A'	0	0	0	0	0	0	0	2
v0B'	1	0	0	0	0	1	0	1
v1B'	0	0	0	0	0	0	0	1
v2B'	0	0	0	0	0	0	0	1
v0C'	0	0	1	0	0	0	0	2
v1C'	0	0	0	0	0	0	0	2
v0A'	0	1	0	0	0	0	0	1
v1A'	1	1	0	0	0	0	0	1
v0B'	0	0	0	1	1	0	0	2
v1B'	0	0	1	1	0	0	0	2
v2B'	0	0	1	0	1	0	0	2
v0C'	0	0	0	0	0	0	1	1
v1C'	0	0	0	0	0	1	1	1
	1	1	2	2	2	1	1	Pop

Figure 2: The input matrix corresponding to the bipartite graph (BMC problem) shown in Figure 1. The population (1 or 2) the molecule belongs to is shown in the rightmost column, and the bottom most row shows the consensus cut in population 1 or 2 in the optimal configuration. Also notice that the solution to the KP problem is such that the molecules/rows and sites/columns corresponding to the vertices of a *gadget* of a vertex of the graph with the MC problem, belong to the same population. The partition suggested by this solution for the MC problem (of Figure 1) is $\{B\}$ and $\{A, C\}$.

For some $\epsilon > 0$, let $\tilde{C}_{KP} \geq (1-\epsilon)C^*_{KP}$.

$$\begin{aligned}
\tilde{C}_{MC} &\geq \tfrac{\tilde{C}_{BMC}}{2} && \text{(eqn (2))} \\
&\geq \tfrac{\tilde{C}_{KP}-6e}{2} && \text{(eqn (3))} \\
&\geq \tfrac{(1-\epsilon)C^*_{KP}-6e}{2} \\
&\geq \tfrac{(1-\epsilon)(C^*_{BMC}+6e)-6e}{2} && \text{(eqn (3))} \\
&\geq \tfrac{(1-\epsilon)C^*_{BMC}-6e\epsilon}{2} \\
&\geq (1-\epsilon)\tfrac{C^*_{BMC}}{2} - 3e\epsilon \\
&\geq (1-\epsilon)C^*_{MC} - (3\epsilon)2C^*_{MC} && (C^*_{MC} \geq e/2) \\
&\geq (1-7\epsilon)C^*_{MC}
\end{aligned}$$

This shows that given a polynomial time approximation scheme (PTAS) for KP, we can construct a PTAS for MC, which is a contradiction; hence KP does not have a PTAS. This concludes the proof of the inapproximability of the KP problem. □

Let $1-\Upsilon$ denote the upper bound on the polynomial time approximation factor of the well-known max cut problem.

Corollary 1 *Achieving an approximation ratio $1-\Upsilon/7$ for the K-populations problem is NP-hard.*

Theorem 2 *There does not exist a polynomial time algorithm (assuming $P \neq NP$) that guarantees the estimation of $(1-\Upsilon/7)p^k_{\max}/p^k_{\min}$ of the total number of consensus cuts in each population (k) where $p^k_{\min}$ and $p^k_{\max}$ are the minimum and maximum of the digestion rates in the population k of the given problem.*

Proof Sketch: For convenience of notation, let us name the problem of maximizing the number of consensus cuts in each population as the $\mathrm{KP}_{\max}$ problem (where a consensus cut, j, in each population k is such that it has at least $p^k_j m_k$ cuts in the position j and m_k is the size of the population k). We will show that if we have a PTAS for $\mathrm{KP}_{\max}$, we will have a PTAS for the k-populations problem, KP, which would be a contradiction. Given a KP, let $p^k_{\min} = \min_j p^k_j$, and $p^k_{\max} = \max_j p^k_j$ for each population k. Let $\tilde{X}$ denote an approximate solution and X^* denote the optimal solution. Then if $\mathrm{KP}_{\max}$ has a PTAS let $\frac{\tilde{N}_k}{N_k} \geq \epsilon$ for some $0 < \epsilon \leq 1$ where N^*_k is the number of consensus cuts in population k. Let $\tilde{C}_k \geq \tilde{N}_k p_{\min}$, then $C^*_k \leq N^*_k p_{\max}$. Hence we have

$$\frac{\tilde{C}_k}{C^*_k} \geq \frac{\tilde{N}_k p_{\min}}{N^*_k p_{\max}} \geq \epsilon \frac{p^k_{\min}}{p^k_{\max}},$$

for each population k. Summing over all the populations, we get a PTAS for the KP problem, which is a contradiction. Thus using corollary 1 we get the required result. □

Corollary 2 *There does not exist a polynomial time algorithm (assuming $P \neq NP$) that guarantees the estimation of $(1-\Upsilon/7)$ of the total number of consensus cuts in each population when the digestion rate at each site is the same.* □

3 A 0.756-approximation algorithm for a 2-populations problem

We give a 0.756-approximation algorithm for a 2-populations problem using the semi-definite programming based algorithm for the Max-cut problem [GW94], on appropriately pruned data. This is similar to the construction presented in [Par97a, Par98]. Again, the constructions require important modifications which have been incorporated in the following discussion.

Given the input, an $m \times n$ binary matrix, we assume we can trim the columns of the input using thresholds λ_1 and λ_2. Every column that has a number of 1's larger than $m\lambda_1$ is a cut in both the populations and is removed. Similarly, every column that has a number of 1's smaller than $m\lambda_2$ is not a cut in either of the populations, and is removed. After the trimming, we are left with a version of the problem where every column is a consensus cut either in population 1 or in population 2. We further assume that the digestion rate for every column is 50%.

We first show that, under these conditions, the 2-populations (2P) problem can be reduced to a *complete* BMC problem and *vice-versa*.

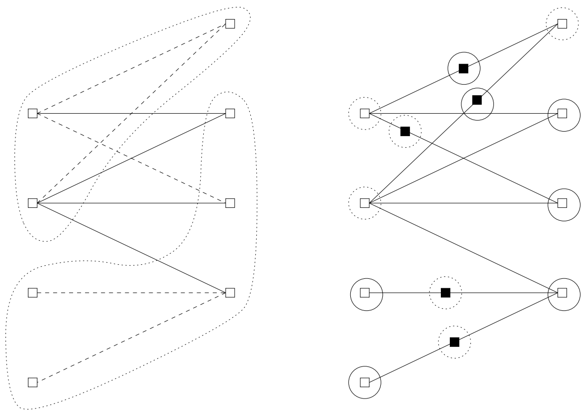


Figure 3: The graph for the BMC is shown on the left where an edge with negative weight is shown as a dashed line. The optimal partition of the vertices is shown in the dotted closed curves enclosing the vertices. The graph for the instance of the MC problem shown on the right is constructed from the one on the left. Every edge with weight -1 has been replaced with two edges and a vertex (shown as solid black rectangle to distinguish it from the other vertices shown as hollow rectangles). The corresponding optimal solution for the MC problem is shown by enclosing each vertex either in a solid circle or a dotted circle; the solid circle enclosed vertices belong to one partition in the solution and the dotted circle enclosed vertices belong to the other.

Given an instance of the 2-populations problem given by an $m \times n$ binary matrix $[a_{ij}]$, we construct a *complete* BMC for a bipartite graph isomorphic to $K_{m,n}$ with vertices v_1^1, v_2^1, $\ldots$, v_m^1 in the first partition and v_1^2, v_2^2, $\ldots$, v_n^2 in the second partition with the weights, $wt(\cdot,\cdot)$, defined as follows:

$$wt(v_i^1, v_j^2) = \begin{cases} 1, & \text{if } a_{ij} = 1; \\ -1, & \text{if } a_{ij} = 0. \end{cases}$$

Notice that given an instance of a complete BMC, we can similarly construct an instance of the 2-populations problem. As a result the following identity must hold:

$$a_{ij} - p_j = wt(v_i^1, v_j^2)/2,$$

since $p_j = 1/2$.

The correspondence between the solutions is as follows. Without loss of generality, let rows (molecules) 1, 2, $\ldots$, m_1 belong to the first population and the remaining rows $m_1 + 1$, $m_1 + 2$, $\ldots$, m belong to the second. Thus $X_{i1} = 1$ for $i \in [1..m_1]$ and $X_{i2} = 1$ for $i \in [m_1 + 1..m]$. Again, without loss of generality, let columns (sites) 1, 2, $\ldots$, n_1 be the consensus cut sites in the first population and the remaining columns $n_1 + 1$, $n_1 + 2$, $\ldots$, n in the second. Thus $Y_{1j} = 1$ for $j \in [1..n_1]$ and $Y_{2j} = 1$ for $j \in [n_1 + 1..n]$. Thus the cost of this partition is simply:

$$\sum_{k=1}^{2} \sum_{j=1}^{n} \sum_{i=1}^{m} X_{ik} Y_{kj} [a_{ij} - p_j]$$

$$= \frac{1}{2}\left(\sum_{j=1}^{n_1}\sum_{i=1}^{m_1} wt(v_i^1, v_j^2) + \sum_{j=n_1+1}^{n}\sum_{i=m_1+1}^{m} wt(v_i^1, v_j^2)\right).$$

The corresponding partition of the BMC problem is then as follows: the vertices of the first partition are $\{v_1^1, v_2^1, \ldots, v_{m_1}^1, v_{n_1+1}^2, v_{n_1+2}^2, \ldots, v_n^2\}$, and the vertices in the second partition are $\{v_{m_1+1}^1, v_{m_1+2}^1, \ldots, v_m^1, v_1^2, v_2^2, \ldots, v_{n_1}^2\}$. Thus, it immediately follows that 2P has a solution of size x iff BMC has a solution of size $2x$. Notice that

$$C_{BMC} = p_1 - l_1, \tag{4}$$

where p_1 is the number of edges with weight 1 and l_1 is the number of edges with weight -1 in the cut. Let l_2 be the remaining number of edges with weight -1 and L be the total number of edges with weight -1. Thus

$$L = l_1 + l_2. \tag{5}$$

Instead of counting only the 1's in the consensus cut columns, let the cost function measure all the "correct decisions" in the configuration. Notice that the optimal is obtained at the same configuration for both the cost functions. Thus, let the cost function of the 2P problem be the number of 1's in the consensus cut columns and the number of -1's in the columns that are not consensus cuts in each of the two populations ($p_1 + l_2$). Thus using equations (4) and (5), we have

$$C_{2P} = L + \frac{C_{BMC}}{2}. \tag{6}$$

Given a BMC, we construct an instance of the MC problem (with weight on the edges as 1) by replacing every edge with a negative weight by two edges and a vertex, each edge having a weight of 1. See Figure 3 for an illustration. If L is the number of edges with weight -1, then the MC instance has $m + n/2 + L$ vertices.

Now, we give the correspondence between the solutions in each of the problem. Notice that the edges introduced in the reduction come in pairs. Let the solution to the MC problem include l_1 edges which are *not* paired, $2l_2$ paired edges and p of the original edges (which had a weight of 1 in the BMC problem). Then

$$C_{MC} = p + l_1 + 2l_2, \tag{7}$$

and the cost of the BMC problem by the construction is,

$$\frac{C_{BMC}}{2} = p - l_1. \tag{8}$$

Since $l_1 + l_2 = L$, we have from equations (7) and (8),

$$\frac{C_{BMC}}{2} = C_{MC} - 2L.$$

From equation (6) we have

$$C_{2P} = C_{MC} - L \tag{9}$$

Finally, we use the algorithm presented in [GW94] to obtain a 0.878-approximation algorithm for the MC problem. Let $\tilde{C}_X$ denote an approximate solution and C_X^* denote the optimal solution to problem X.

$$\begin{array}{rcll}
\tilde{C}_{2P} & = & \tilde{C}_{MC} - L & \text{(from eqn (9))} \\
& \geq & 0.878 C_{MC}^* - L & \text{(from [GW94])} \\
& \geq & 0.878 (C_{2P}^* + L) - L & \text{(from eqn (9))} \\
& = & 0.878 C_{2P}^* - 0.122 L & \\
& \geq & 0.756 C_{2P}^* & \text{(from eqn (6))}
\end{array}$$

This concludes the argument.

4 A Probabilistic Analysis

We make use of the following notation:

# of populations = K
# of molecules per population = m'
# of molecules over all populations = $m = Km'$
# of singleton cuts per population = k
# of non-singleton cuts over all populations = k'
# of cuts over all populations = $Q = Kk + k'$, and
Digestion rate = p.

In this analysis we shall assume that there is no sizing error and false cut rate $q = 0$. Note that a cut site is called a "singleton" if that site belongs to one population; otherwise, it's a "non-singleton" cut site with a multiplicity of $\alpha \geq 2$ (i.e., it appears in α different populations). Note that the number of cuts in a map for any single population is bounded by $k + k'$.

The analysis technique is very similar to the one provided in [AM98], but requires some important modifications. Many of the details are intentionally omitted as they can be inferred from the earlier analysis applied to the single population case.

4.1 False Negative Errors: Partial Digestion

Let us postulate an experiment, where the desired *normalized ordered restriction map* is observed, subject to *partial digestion* error where any particular restriction site is observed with some probability $p \leq 1$. We assume no other error sources; thus no other spurious sites (false restriction cuts) are included in the observation; the observed restriction map appears in the correct orientation; and there is no sizing error.

We claim that if

$$m > \frac{K}{p}[c + \ln(Kk + k')]$$

(where $k \geq 1$ and $c \geq 1$) then the result of a straightforward algorithm that simply includes every observed cut site is correct with probability greater than $e^{-e^{-c}} e^{-(e^{-2c})/2Q}$. Note that the probability that a cut site does not appear in any particular observation is $(1-p)$ and thus the probability that it does not appear in any of the m' independent observations corresponding to its population is $(1-p)^{m'}$. Thus, we see that the probability that every true cut site appears in the final result is $[1-(1-p)^{m'}]$. Note that

$$(1-p)^{m'} \leq e^{-pm/K} \leq e^{-c-\ln Q} = \frac{e^{-c}}{Q},$$

and

$$1-(1-p)^{m'} \geq 1 - \frac{e^{-c}}{Q}.$$

Thus the probability that all Q true cut sites show up in the final map is given by

$$[1-(1-p)^{m'}]^Q > e^{-e^{-c}}\ e^{-(e^{-2c})/2Q}.$$

4.2 Eliminating Non-singleton Cuts

In the next phase, we shall identify a cut site to be a non-singleton cut, if it appears in more than $\frac{10}{7}m'p$ molecules. Assume that

$$m > \frac{49K}{p} \max\left[\frac{c+\ln k'}{4}, \frac{c + \ln k + \ln K}{3}\right].$$

Note that if a cut site is a "singleton," then the probability that it will be accidentally labeled as a non-singleton is bounded by the following application of Chernoff's bound [ASE92]:

$$\begin{aligned} & Pr\left[S(m',p) \geq \left(1+\frac{3}{7}\right) m'p\right] \\ \leq\ & e^{-(3/49)m'p} < e^{-c-\ln(Kk)} = \frac{e^{-c}}{Kk}. \end{aligned}$$

Thus the probability that none of the singleton cut site will be mislabeled is bounded by

$$\left(1-\frac{e^{-c}}{Kk}\right)^{Kk} > e^{-e^{-c}}\ e^{-(e^{-2c})/2Kk}.$$

Now in the other direction, we see that a non-singleton cut site with a multiplicity $\alpha \geq 2$ will go undetected is bounded by the following application of Chernoff's bound [ASE92]:

$$\begin{aligned} & Pr\left[S(\alpha m',p) \leq \left(1-\frac{2}{7}\right) 2m'p\right] \\ \leq\ & e^{-(4/49)m'p} < e^{-c-\ln(k')} = \frac{e^{-c}}{k'}. \end{aligned}$$

Thus the probability that all of the non-singleton cut sites will be correctly labeled is bounded by

$$\left(1 - \frac{e^{-c}}{k'}\right)^{k'} > e^{-e^{-c}} e^{-(e^{-2c})/2k'}.$$

4.3 Grouping Singleton Cuts

In the third phase, we will group the singleton cuts in to K groups where the i^{th} group contains those and only those singleton cuts that belong to one distinct population. In order to do this, we proceed as follows: Define a graph $G = (V, E)$, where $V = \{v_1, v_2, \ldots, v_{Kk}\}$ corresponds to the singleton cuts and $e = [v_i, v_j] \in E$ if and only if the singleton cuts associated with v_i and v_j appear in one single observation.

Assume that

$$m > \frac{K}{p^2} \ln\left(\frac{k}{k - \ln k - c}\right).$$

It is easy to see that if v_i and v_j are not two singleton cuts belonging to the same population, then there can be no edge between them. On the other hand, if in fact they do belong to the same population, then the probability that this edge does not occur is

$$(1-p^2)^{m'} \leq e^{-p^2 m'} \leq e^{-\ln(k/k - \ln k - c)} = \frac{k - \ln k - c}{k} = 1 - \left(\frac{\ln k}{k} + \frac{c}{k}\right),$$

and the edge probability is

$$p_e = 1 - (1-p^2)^n \geq \frac{\ln k}{k} + \frac{c}{k}.$$

Thus by the well-known result on the connectivity in random graphs [Spe87], we see that with $p_e \geq \frac{\ln k}{k} + \frac{c}{k}$, and the subgraph induced by the k singleton cut vertices belonging to the same population is connected with probability $e^{-e^{-c}}$, in the limit as $k \to \infty$.

Thus almost surely, we will be able to group the singleton cuts.

4.4 Grouping Non-singleton Cuts

Finally, we need to associate each non-singleton cut to all the populations that it belongs to. Consider a particular non-singleton cut and an arbitrary observation from the population containing the non-singleton cut. If that particular observation contains the non-singleton cut and *at least* one non-singleton cut, then we can assign the non-singleton cut site to the correct population.

Assume that

$$m > \frac{K}{p}\left(\frac{c + \ln k' + \ln K}{(1 - e^{-pk})}\right).$$

Note that the probability of failure for any given observation is

$$(1-p) + p(1-p)^k \leq 1 - p(1 - e^{-pk}).$$

The probability that this occurs for all of the m' observations in that population is bounded by:

$$(1 - p(1 - e^{-pk}))^{m'} < e^{-m'p(1-e^{-pk})} < e^{-c-\ln Kk'} = \frac{e^{-c}}{Kk'}.$$

As a result the probability that we can group all the non-singleton cuts with the singleton cuts belonging to a particular population is given by

$$\left(1 - \frac{e^{-c}}{Kk'}\right)^{k'}.$$

Thus the probability that the non-singleton grouping can be done for all the populations correctly is bounded by

$$\left(1 - \frac{e^{-c}}{Kk'}\right)^{Kk'} > e^{-e^{-c}}\, e^{-(e^{-2c})/2Kk'}.$$

At this point we have all the population based maps, and it is straightforward to partition the complete set observations into K classes. Note, however, that certain observations (roughly me^{-pk} observations) containing only non-singleton cut sites cannot be unambiguously associated with any one distinct population. However, this does not pose a serious problem as any straightforward rule for disambiguation leaves the computed individual maps unaffected.

In summary, we have

Theorem 3 *Let ϵ be a positive constant and $c \geq 1$ be so chosen that $1 - e^{-(K+4)e^{-c}} = \epsilon$. Then for*

$$m > \frac{49K(1+e^{-c})}{3p} \max\left[c + \ln(Kk + k'), \frac{c + \ln Kk'}{1 - e^{-pk}}, \frac{1}{p} \ln\left(\frac{k}{k - \ln k - c}\right)\right],$$

($k > c + \ln k$), with probability at least $1 - \epsilon$, the correct ordered restriction map can be computed in $O(m(k^2 + Kk + k'))$ time. □

5 An algorithm for the K-populations problem

Here we present a set of heuristics to detect the different populations in a sample of molecules. In our experiments, we have modeled the false positive, false negative, orientation and sizing errors in the input. Let

p be the digestion rate (true positive),

q be the false positive rate,

m be the sample size, and,

$m_{\min}$ be the smallest sample size for any population.

Thus

$$m_{\min} K \leq m.$$

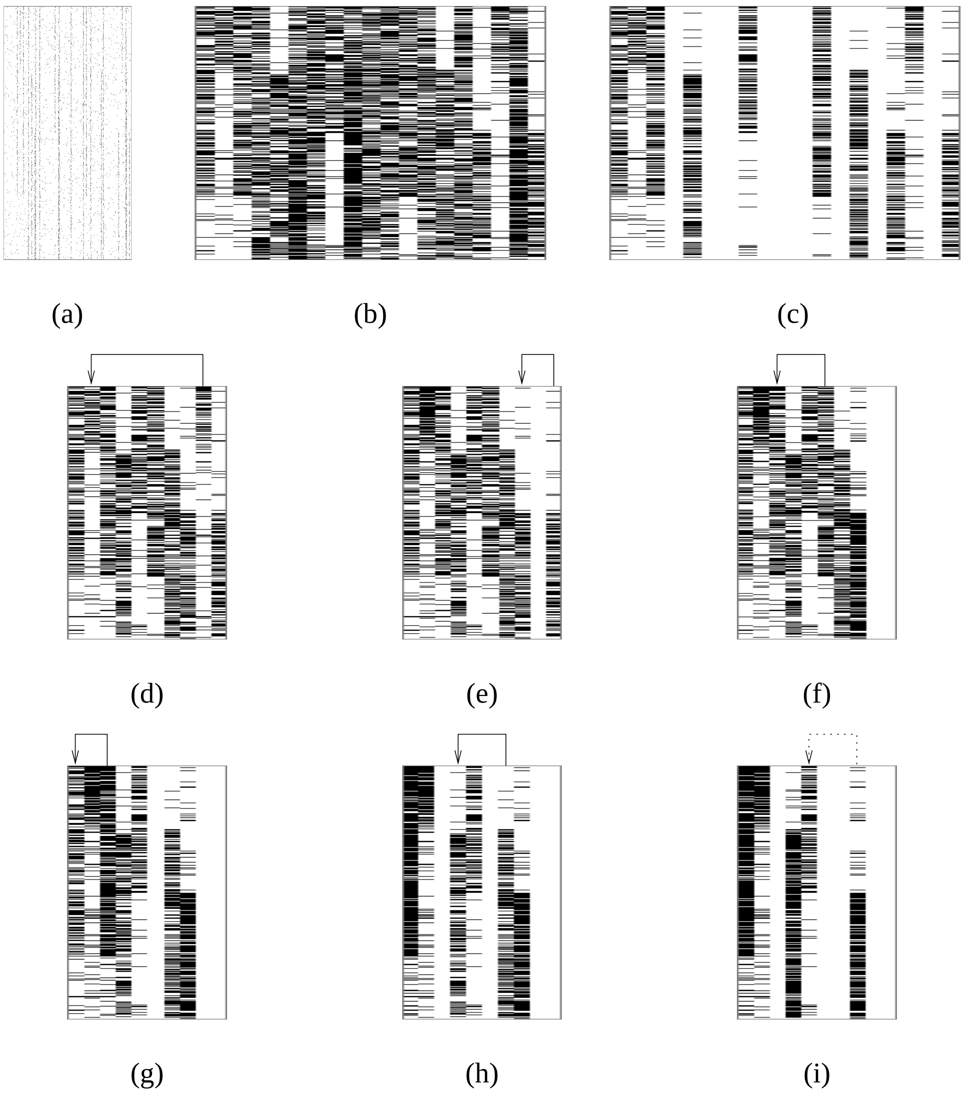


Figure 4: Illustration of step 2 (merging and error correction): (a) The correctly aligned molecules with the computed consensus cuts. In (b) to (i), each cut is shown by a small horizontal bar. (b) This shows only the consensus cuts in the population aligned in (a). (c) For the very first merging, all those cuts that appear in *all* the populations have been filtered. (d) to (i) show each step of the merge; the polyline on top denotes the two columns with identical *population map* being merged. Notice the "darkening" of the merged column due to the error correction. In (i) the two columns to be merged actually have population maps that are complements of each other.

The population detection is carried out in three major steps: First we detect the physical map which is the union of the map of all the populations. Next, we merge identical consensus cuts (characterized by their presence or absence in a population, defined by the *population map*). At this step we also carry out a certain amount of error correction which makes the next step more robust. In the final step, we use the merged consensus cut columns of the last step to separate the populations. We give the details of each step below.

(Step 1) Common physical map detection. Obtain the physical map, which is the union of the physical maps of all the populations. This step eliminates the orientation, sizing and false positive errors. Here we must assume that the false positive rate is low enough not to be confused as a true positive of a "small" population; thus the following must hold:

$$mq \ll m_{\min} p \quad \Rightarrow \quad Kq \leq \frac{m}{m_{\min}} q \ll p. \tag{10}$$

In our experiments, we use the EBFC-based algorithm (Exclusive Binary Flip Cut) presented in [MP96].

(Step 2) Merging consensus cuts and error correction. Given a consensus cut j, define a *population map* for j as follows: let the number of populations be K, then the population map is a K length vector of 0's and 1's, where a 1 at position i denotes j is a consensus cut in population i and a 0 denotes j is *not* a consensus cut in population i.

Let $j_{pm_1}, j_{pm_2}, \ldots, j_{pm_l}$ be l cuts that have the same population map. We replace all these l consensus cut columns by a single *representative* column. In addition, we also carry out false negative error correction in the merged (representative) column as described below.

(Step 2.1) Identifying the consensus cuts for merging. Recall that at this stage of the algorithm we do not know what the populations are (nor how many of them). We will identify the consensus cuts with identical population maps by simultaneously looking at c consensus cuts for the presence of at least t pairwise true positives. The reader may verify that in the worst case,

$$c \geq \lceil \tfrac{1}{p} \rceil \quad \text{and} \quad t = m_{\min} p.$$

Thus for $p \geq 0.5$, it suffices to consider *pairwise* cuts (i.e., $c = 2$). The merging is carried out iteratively until no more columns can be merged. A weight wt is associated with every merged column which is the number of consensus cut columns that were merged.

(Step 2.2) Error (false negative) correction. We give the details for the case where $c = 2$, the other cases are similar. Once we recognize that two consensus cuts j_1 and j_2 have the same population map, it is easy to make the error correction: if j_1 and j_2 do not agree in molecule i, we place a consensus cut for molecule i in the merged column. Assuming that the population map of j_1 and j_2 is correct, this consensus cut in molecule i is correct with probability p and wrong with probability q (recall that $q << p$). See Figure 4 for an example.

(Step 3) Separating the populations. Let the number of representative columns be n_r. The reader may verify that

$$\log K \leq n_r \leq n.$$

Routine <u>Separate-Populations</u>

Set $Tol = pm_{\min}$.
Initialize every molecule, i, to be in population p_0.
For each column, j, in the descending order of wt do {
  For each population, p, with size $> Tol$, do {
    Compute N_0^{jp}, the number of molecules with 0 at j.
    Compute N_1^{jp}, the number of molecules with 1 at j.
    If ($N_0^{jp} > Tol$ and $N_1^{jp} > Tol$) {
      Split the population by the 0's and 1's.
      Renumber the new populations as p' and p''.
    }
  }
}

A *similarity* tree.

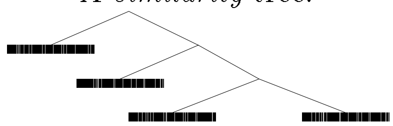


Figure 5: The algorithm to separate the populations using the representative columns, and an example of a tree based on the decisions taken at the "if" statement. The physical map of each population is at the leaf node of this tree.

The pseudocode for the algorithm to detect the different populations is presented in Figure 5. This algorithm splits the molecules into different populations and also gives a tree, based on the iterative splitting of the sample, to give rise to a *similarity tree*. The common structure of the maps of the different populations is captured in this tree.

6 Experimental Results

This section has two parts. In the first part, we use the algorithm presented in the last section to obtain the population map of upto 6 populations. In the second part, we experimentally verify the smallest number of molecules required to compute the correct population maps.

Part I: We carried out several experiments using simulated data on upto six populations data. We present the results of a two, a four and a six population sample in Figures 6 through 10. We observed in our experiments that as the number of populations increase, we need to increase the digestion rate to detect the different populations. In the figure each molecule is represented by a row with a black dot denoting a cut in the molecule. For ease of visualization of the output, in the synthesized data, the molecules from each population are placed close to one another (the algorithm neither knows nor uses this information). As can be seen, the algorithm separates the populations with about 20% error (picking a wrong molecule or missing out a molecule), but gathers sufficient information to infer the right population maps.

Part II: We also use the algorithm of the last section to check the number of molecules required to correctly compute each of the population maps. In our experiments we have considered two populations. For each data set and a fixed number of molecules, we generated 50 random data samples and plotted the fraction of the data sets solved correctly against the number of molecules in the data set. Here we assume that either the algorithm gives the two population maps correctly or it does not. Ideally, this could be a value between 0 and 1, but for our purposes we assign a (non-forgiving) value of 0 to any map that errs even on a single cut. As can be seen in Figures 11, 12 and 13, there is a cut-off point beyond which the algorithm always gives the correct populations maps which is in agreement with the calculations in Section 4. In all the experiments we used a digestion rate of 0.2, no false positive, sizing and orientation errors. The size of the two populations are roughly (though not exactly) equal in all the data sets.

7 Conclusion

We have presented our initial complexity analysis and some heuristics for identifying K ($K \geq 2$) distinct populations from a set of corrupted ordered restriction map data. Our main results are the following: (1) The problem (even under a forgiving idealized model) is computationally infeasible. (2) We give a guaranteed 0.756-approximation algorithm for a special class of the 2-populations problem. (3) We give a probabilistic analysis of the model along with experimental verification. (4) However, for the general K-populations problem, when certain assumptions are made regarding the errors in the input data (e.g., false positive error probability is negligible compared to false negative error probability), we have been able to empirically obtain very promising results. It remains open whether using better prior models of the data as well as variations on the heuristics presented here we can devise algorithms with better performance for this problem while accounting for many other error sources.

Acknowledgment

Our thanks go to Rohit Parikh, Raghu Varadhan, Joel Spencer, Thomas Anantharaman and David Schwartz for many helpful comments and encouragement.

References

[ASE92] N. Alon, J.H. Spencer and P. Erdös, *The Probabilistic Method*, Wiley Interscience, John Wiley & Sons, Inc., NY, 1992.

[AMS97] T.S. Anantharaman, B. Mishra and D.C. Schwartz, "Genomics via Optical Mapping II: Ordered Restriction Maps," *Journal of Computational Biology*,**4**(2):91–118, 1997.

[AM98] T.S. Anantharaman and B. Mishra, "Genomics via Optical Mapping I: Probabilistic Analysis of Optical Mapping Models," Courant TR, 1997/1998.

[AL97] S. Arora and C. Lund, "Hardness of Approximations," *Approximation algorithms for NP-Hardness Problems*, (Ed. D.S. Hochbaum), PWS Publishing Company, MA, 1997.

[Cai+95] W. Cai et al., "Ordered Restriction Endonuclease Maps of Yeast Artificial Chromosomes Created by Optical Mapping on Surfaces," *Proc. Natl. Acad. Sci., USA*, **92**:5164–5168, 1995.

[DHM97] V. Dančík, S. Hannehalli and S. Muthukrishnan, "Hardness of Flip-Cut Problems for Optical Mapping," *Journal of Computational Biology*,**4**(2), 1997.

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

[GP96] D. Geiger, L. Parida, *A Model and Solution to the DNA Flipping String Problem*, Courant Inst. of Math. Sciences, New York University, TR1996-720, May, 1996.

[GW94] M. X. Goemans, D. P. Williamson, ".878-approximation algorithms for MAX CUT and MAX 2SAT", *Proceedings of the Twenty-Sixth Annual ACM Symposium on Theory of Computing*, pp 422-431, Montreal, Quebec, Canada, 23-25 May 1994.

[KS98] R. Karp and R. Shamir, "Algorithms for Optical Mapping," In *Proceedings 2nd Annual Conference on Computational Molecular Biology*, (RECOMB '98), ACM Press, 1998.

[Men+95] X. Meng et al., "Optical Mapping of Lambda Bacteriophage Clones Using Restriction Endonuclease," *Nature Genetics*, **9**:432–438, 1995.

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

[Par97a] L. Parida, "Inapproximability of Flip-Cut and Other Problems for Optical Mapping," submitted for publication. (Also Courant Inst. of Math. Sciences, New York University, TR1997-740, August, 1997.)

[Par97b] L. Parida, "A Uniform Framework for Ordered Restriction Map Problems", to appear in *Journal of Computational Biology*. (Also Courant Inst. of Math. Sciences, New York University, TR1997-739, August, 1997.)

[Par98] L. Parida. *Algorithmic Techniques in Computational Genomics*. PhD thesis, Courant Institute of Mathematical Sciences, New York University, September 1998.

[PM98] L. Parida and B. Mishra. "Partitioning K clones: Hardness results and practical algorithms for the K-populations problem", In *Proceedings of the Second Annual Conference on Computational Molecular Biology (RECOMB98)*, pages 192–201. ACM Press, 1998.

[Sam+95] A. Samad et al., "Mapping the Genome One Molecule At a Time—Optical Mapping," *Nature*, **378**:516–517, 1995.

[Sch+93] D.C. Schwartz et al., "Ordered Restriction Maps of *Saccharomyces cerevisiae* Chromosomes Constructed by Optical Mapping," *Science*, **262**:110–114, 1993.

[Spe87] J. Spencer, *Ten Lectures on the Probabilistic Method*, Society for Industrial and Applied Mathematics, Philadelphia, PA, 1987.

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

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

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

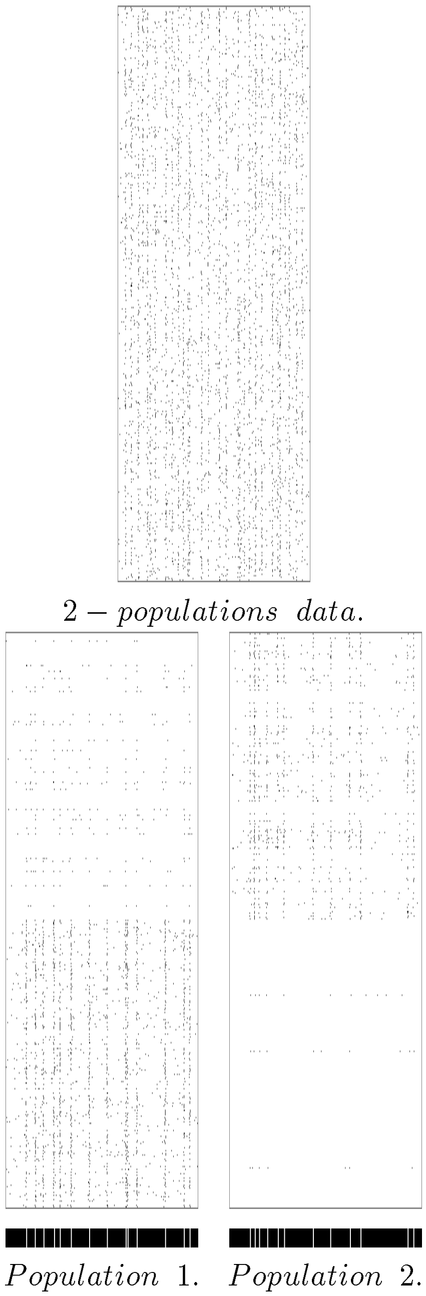
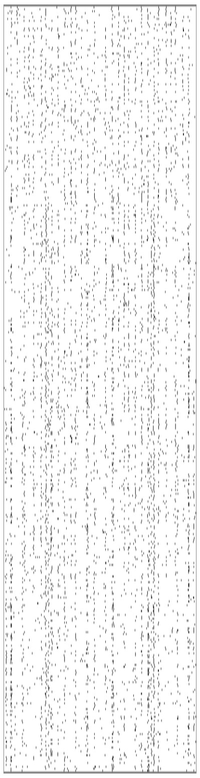


Figure 6: An example of a 2-populations problem. The false negative rate is 50%; the false positive rate is 5%; the orientation is incorrect 50% of the time; small sizing error; the number of consensus cuts is around 15 for each population and the the maps of each population are very similar. The second row shows the aligned molecules of each population (as detected by the algorithm) and the computed physical maps. The algorithm makes an error of about 17% in assigning population to each molecule, but picks up the right map for each population as shown.



$4 - populations\ \ data.$

Figure 7: An example of a 4-populations problem. The false negative rate is 50%; the false positive rate is 5%; the orientation is incorrect 50% of the time; small sizing error; the number of consensus cuts is around 10-15 for each population and the the maps of each population are very similar (as can be seen in the next figure).

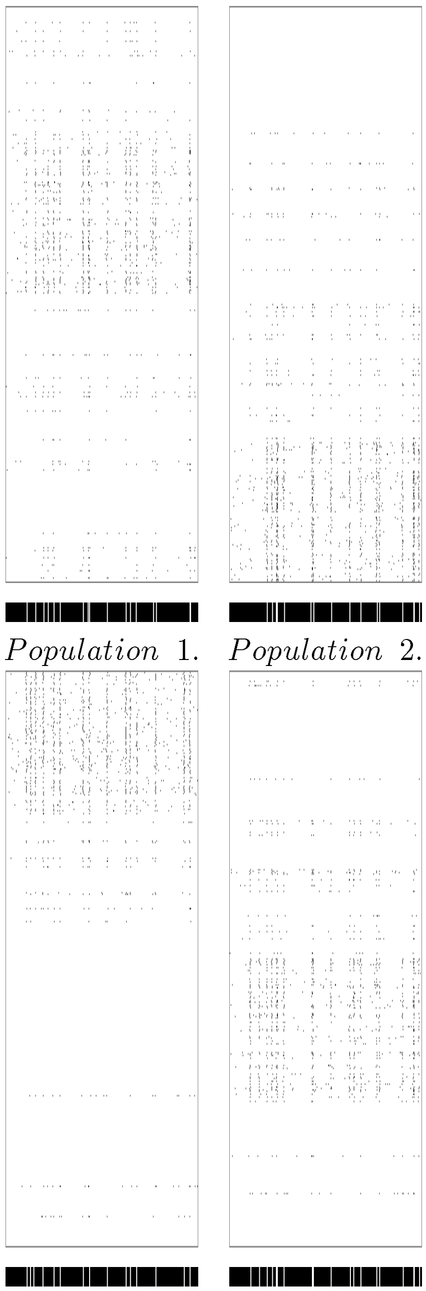
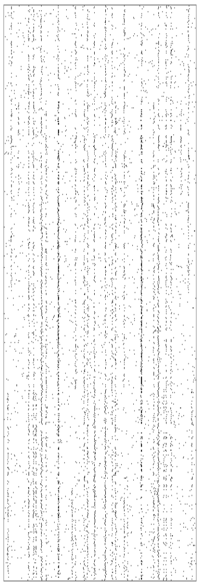
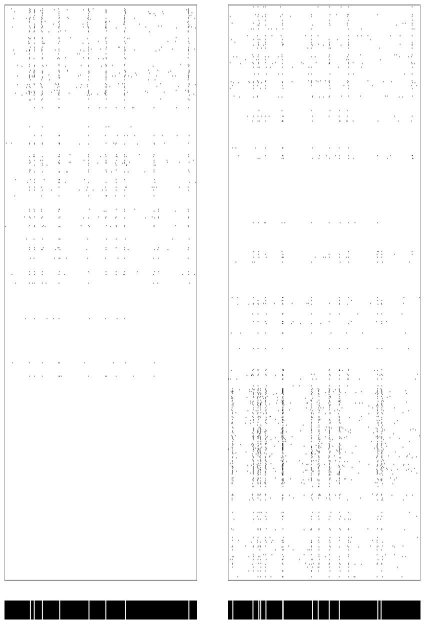


Figure 8: The 4-populations Problem: The algorithm makes an error of about 20% in assigning population to each molecule, but picks up the right map for each population as shown.

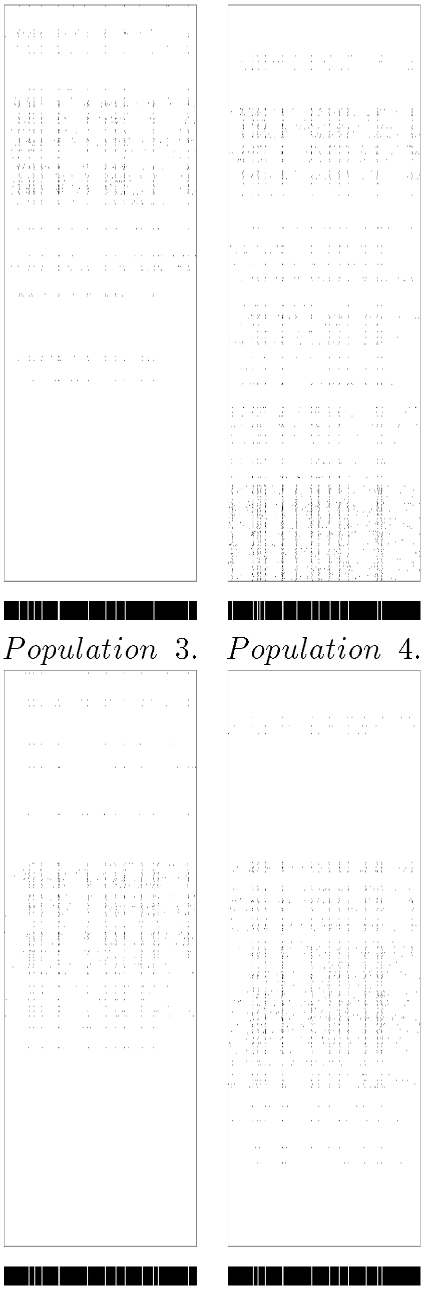


$6 - populations\ \ data.$



$Population\ 1.\quad Population\ 2.$

Figure 9: An example of a 6-populations problem. The false negative rate is 30%; the false positive rate is 5%; the orientation is incorrect 50% of the time; small sizing error; the number of consensus cuts is around 10-15 for each population and the the maps of each population are very similar. We also show two of the populations and their physical maps that the algorithm detects, in the second row (the remaining four are shown in the next figure).



Population 3. *Population* 4.

Population 5. *Population* 6.

Figure 10: The 6-populations Problem: Four of the six populations detected by the algorithm are shown here (the other two are shown in the previous figure). The algorithm makes an error of about 20% in assigning population to each molecule, but picks up the right map for each population as shown.

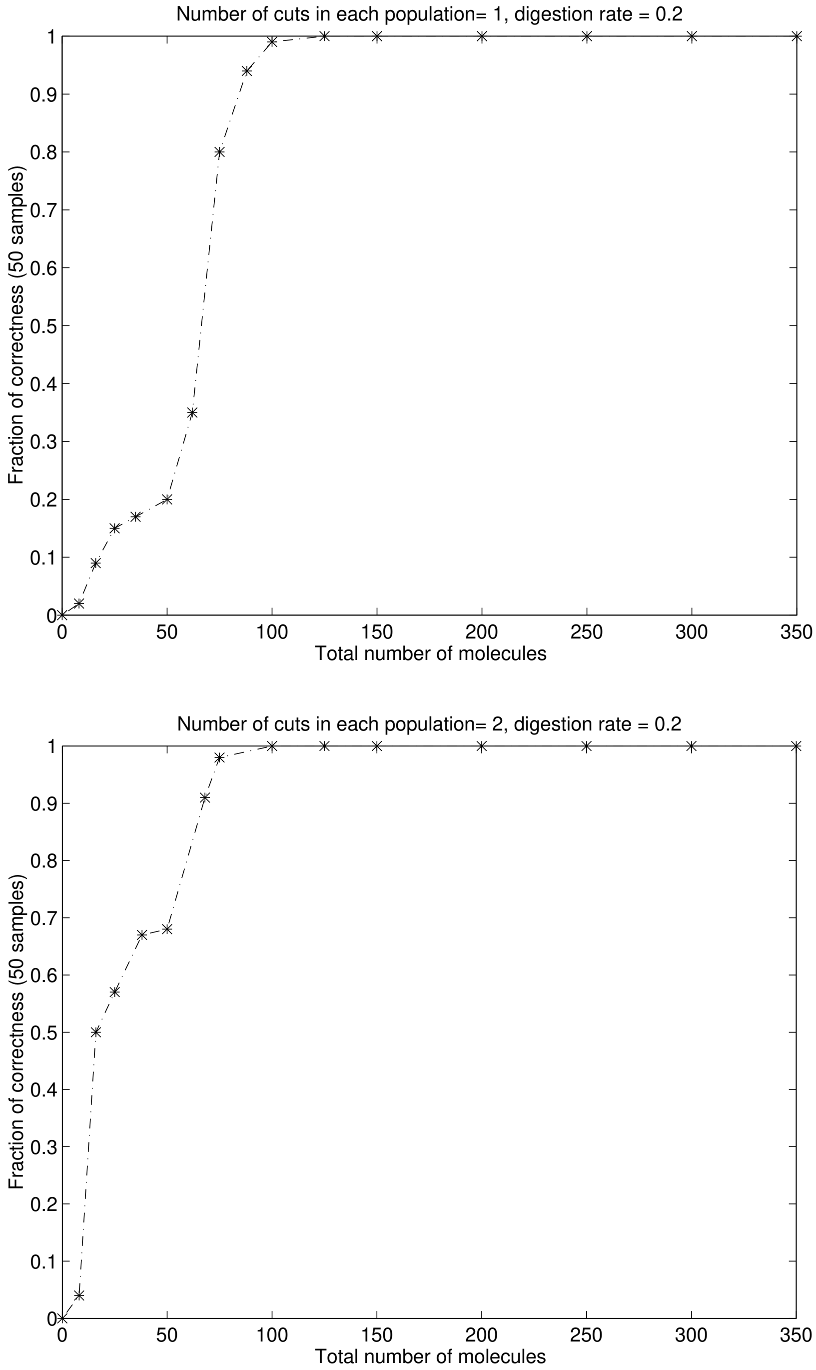


Figure 11: Experimental results using number of cuts in each population as 1 and 2 respectively.

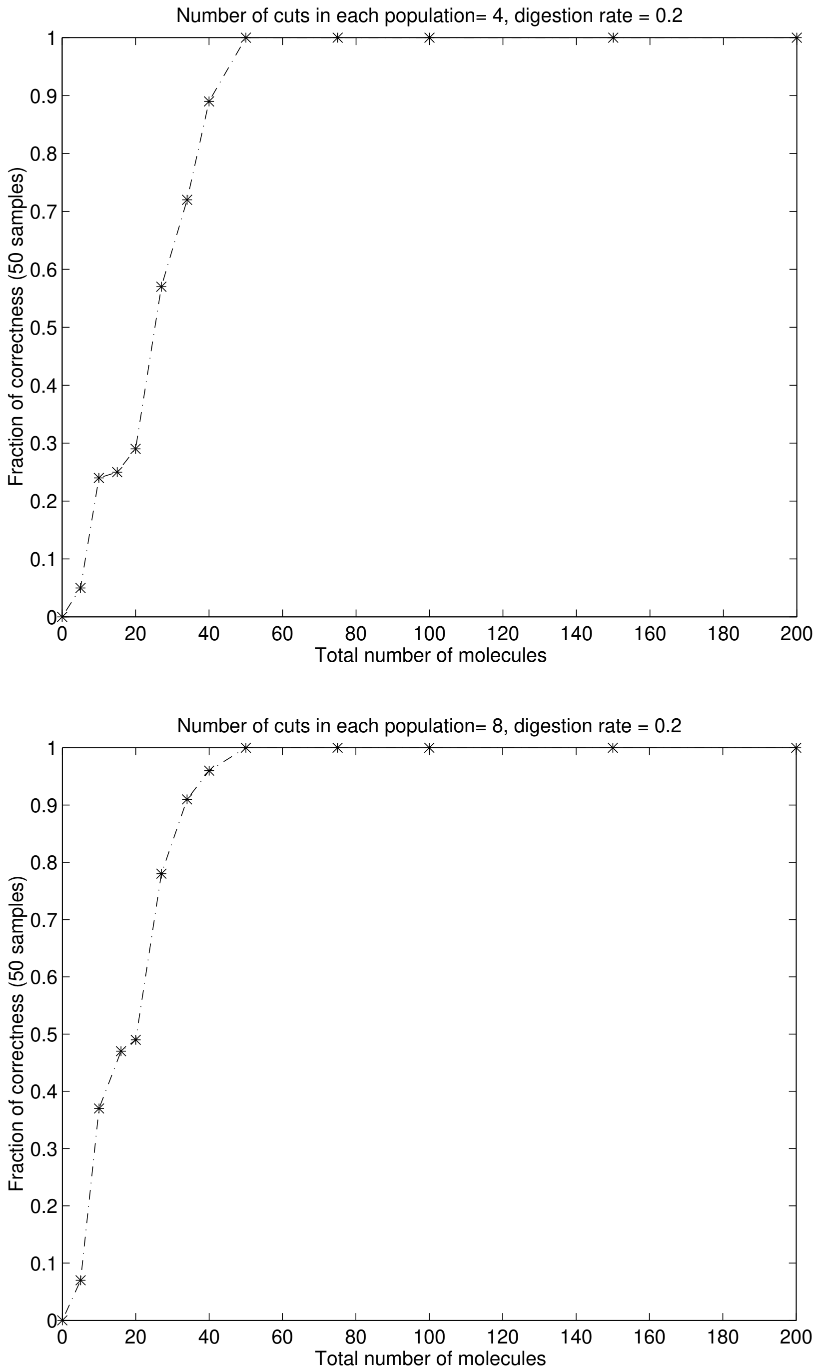


Figure 12: Experimental results using number of cuts in each population as 4 and 8 respectively.

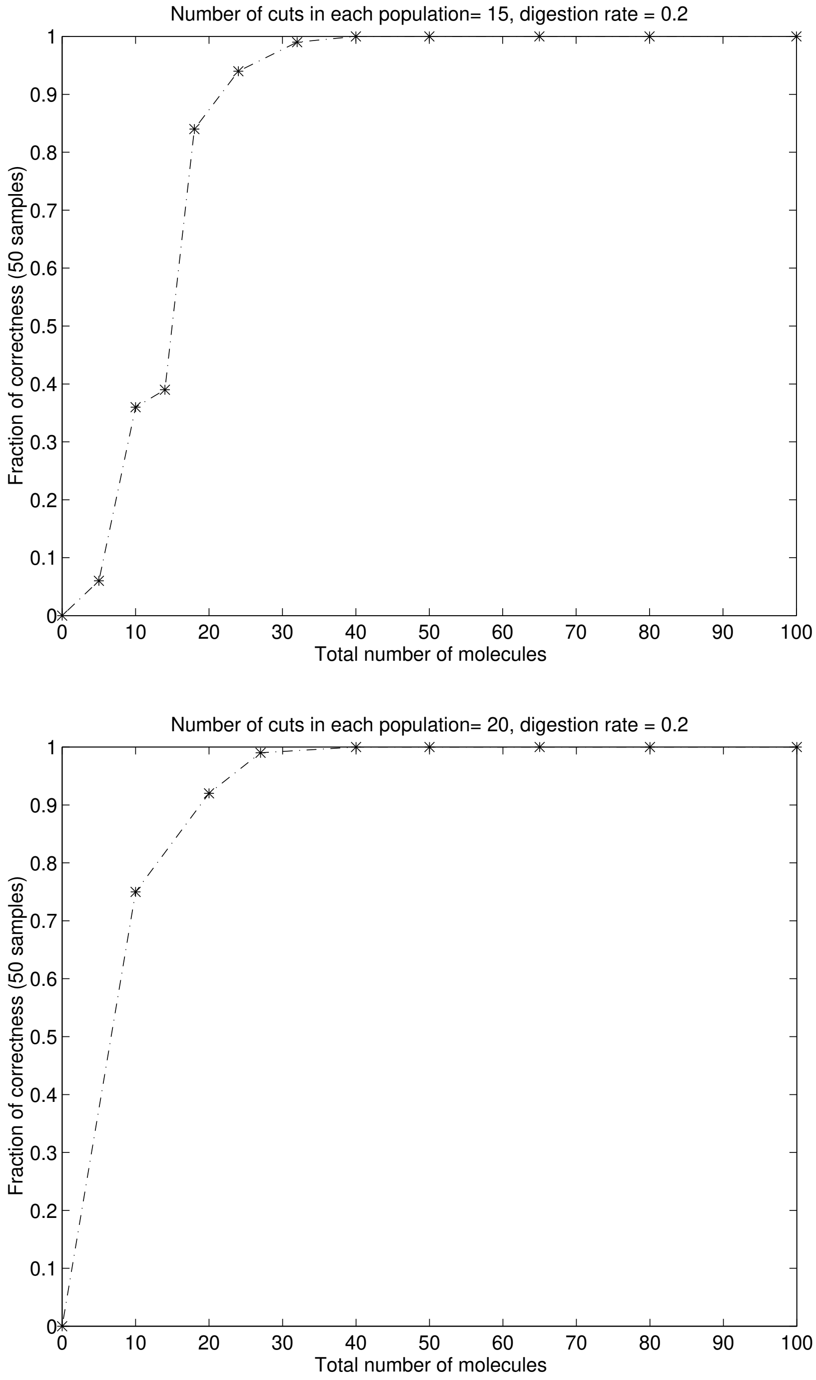


Figure 13: Experimental results using number of cuts in each population as 15 and 20 respectively.