INTERNATIONAL COMPUTER SCIENCE INSTITUTE

1947 Center Street • Suite 600 • Berkeley, California 94704 • 1-510-642-4274 • FAX 1-510-643-7684

Physical Mapping of Chromosomes: A Combinatorial Problem in Molecular Biology

Farid Alizadeh* Richard M. Karp*
Lee A. Newberg* Deborah K. Weisser*

TR-92-066

September 29, 1992

Abstract

A fundamental tool for exploring the structure of a long DNA sequence is to construct a "library" consisting of many cloned fragments of the sequence. Each fragment can be replicated indefinitely and then "fingerprinted" to obtain partial information about its structure. A common type of fingerprinting is restriction fingerprinting, in which an enzyme called a restriction nuclease cleaves the fragment wherever a particular short sequence of nucleotides (letters 'A', 'G', 'C', and 'T') occurs, and the lengths of the resulting pieces are measured. An important combinatorial problem is to determine, from such fingerprint information, the most probable arrangement of the cloned fragments along the overall sequence. However, for a given arrangement, even the likelihood function involves a complicated multifold integral and therefore difficult to compute. We propose an approximation to the likelihood function and develop local search algorithms based on this approximate objective function. Our local search techniques are extensions of similar strategies for the traveling salesman problem. We provide some computational results which support our choice of objective function. We also briefly study alternative approaches based on pairwise probabilities that two fragments overlap.

*Research supported by NSF Grant Nos. CCR-9005448 and CDA-9211106. FA and RMK are at the International Computer Science Institute, Berkeley, CA. RMK, LAN, and DKW are at the University of California, Berkeley.

1 Introduction

A DNA molecule such as a chromosome is a very long polymer consisting of two intertwined strands, each of which is a sequence of *nucleotides* from the set $\{A, T, C, G\}$. The nucleotides A and T are *complementary* to each other, as are the nucleotides C and G. Each nucleotide on one strand is bound to a complementary nucleotide on the other strand. The human genome project and many other efforts in molecular biology aim to sequence the chromosomes of humans and other species and to elucidate the genetic information contained in these sequences.

In order to study a large DNA molecule it is necessary to break it into smaller pieces, study the structure of each piece, and then mathematically reassemble the pieces to determine the structure of the entire molecule. This paper is concerned with algorithms for the reassembly process.

Restriction enzymes can be used to cut a long DNA molecule into fragments. Under appropriate experimental conditions a given restriction enzyme will cleave a DNA molecule at every site where a certain short sequence of nucleotides occurs. A separation process called *electrophoresis* can be used to separate the resulting fragments of the DNA molecule and measure their sizes.

The process of *cloning* incorporates a fragment of a DNA molecule into some self-replicating genetic host. The self-replication process then creates enormous numbers of copies of the fragment, thus enabling its structure to be investigated. A fragment reproduced in this way is called a *clone*.

A *clone library* consists of hundreds or thousands of clones from the same DNA molecule. Typically, the DNA molecule will be 10^7 or 10^8 nucleotides long and each clone will be between approximately $40,000$ and $1,000,000$ nucleotides long, depending upon the type of self-replicating host used. The *coverage* of a clone library is the ratio between the sum of the lengths of the clones and the length of the DNA molecule. Typically, the library provides a coverage in the range 5–10.

Experiments on a given clone produce a partial description called a *fingerprint* of the clone. The fingerprint typically includes two kinds of data: *restriction fragment data* and *hybridization data*. Restriction fragment data is obtained by exposing the clone to a restriction enzyme and measuring the sizes of the fragments produced. Hybridization data determines whether a short molecule called an *oligonucleotide* or *probe*, consisting of about ten nucleotides, binds to the clone. Under the right conditions binding will occur if and only if the clone contains the sequence complementary to the probe.

Our purpose is to reconstruct the relative placement of the clones along the DNA molecule; this information has been lost in the construction of the clone library. In this article we restrict ourselves to the noiseless case, in which the data is free of experimental error. We also restrict ourselves to hybridization data. However, many of our algorithmic techniques can be extended to incorporate noise and restriction fragment data. The experimental points discussed in [CNHZL90] correspond closely to the problem we have formulated.

We assume that the data for the problem consists of a $n \times m$ matrix $D = (d_{ij})$, where

$d_{ij} = 1$ if clone C_i contains probe P_j, and $d_{ij} = 0$ otherwise. Note that the data does not indicate how many times a probe occurs on a given clone, nor does it give the order of occurrence of the probes along the clone. Thus our problem is quite different from the well-known shortest superstring problem. ([BJLTY91, Tur89]).

Our problem may be understood in terms of the following analogy. Several copies of a long newspaper article are cut into pieces. For each piece, and each of a list of key words, we are told whether the piece contains the key word. Given this data, we wish to determine the most likely pattern of overlap of the pieces.

Define a *placement* as an assignment of a closed interval on the real line to each clone. Define an *interleaving* as a specification of the linear ordering of the $2n$ endpoints of these n intervals. We may view an interleaving as an equivalence class of placements with a common topological structure. Given the data, we want to determine the most likely interleaving. To give precise meaning to "most likely" we require a stochastic model.

2 A Stochastic Model

To define the *a priori* probability of a placement or of an interleaving we use the following stochastic model, due to Lander and Waterman. ([LW88]).

1. Each clone is an interval of length 1 contained in the interval $[0, N]$;
2. The left-hand end points of the clones are independent random variables, each of which has the uniform distribution over $[0, N-1]$;
3. For $i = 1, 2, \ldots, m$, the occurrences of probe i along the interval $[0, N]$ are according to a Poisson process of rate λ. These m Poisson processes are mutually independent.

This simple stochastic model does not account for some phenomena, such as the fact that certain long sequences of nucleotides tend to occur repeatedly, but no decisively better model has been proposed.

3 The *a Posteriori* Probability of an Interleaving

Let $p(I|D)$ be the probability of interleaving I given data D. We define the physical mapping problem as that of finding an interleaving that maximizes $p(I|D)$. In practice, it is more useful to provide several interleavings that approximately maximize $p(I|D)$. These can serve as competing hypotheses to be confirmed or disconfirmed by further experiments. Our algorithms are capable of providing such a set of alternate solutions.

A placement can be specified by a point $x = (x_1, x_2, \ldots, x_n)$ in n-dimensional Euclidean space, where x_i denotes the left-hand end point of clone C_i, and $0 \leq x_i \leq N-1$. Let $p(D|x)$ be the *a priori* probability of data D given placement x.

Each placement x defines a linear ordering of the clones, according to the order in which their starting points occur along the real line. With respect to a particular probe A, define

a *positive run* as a maximal sequence of consecutive clones, all of which are incident with probe A, and a *negative run* as a maximal sequence of consecutive clones, none of which are incident with probe A. Then $p(D|x)$ is a product of factors corresponding to the positive and negative runs for all the probes. The factor corresponding to a negative run is the probability that none of the clones in that run are incident with the probe; the factor is simply $e^{-\lambda L}$, where L is the length of the run. For a positive run the factor is the probability that all of the clones in the run contain the probe, given that none of the clones in the adjacent negative runs contain the probe.

For a given placement, define an *atomic interval* as a maximal interval on the real line which contains no end point of a clone in its interior. Define the *height* of an atomic interval as the number of clones containing it. An atomic interval of height zero is called a *gap*. Define the *size* of a placement as the sum of the heights of its atomic intervals.

An interleaving I can be viewed as an equivalence class consisting of those placements for which the $2n$ left- and right-hand end points occur in a specified order. The condition on the order of the end points can be expressed by linear inequalities; for example, the inequality $x_i \leq x_j + 1$ expresses the constraint that the left-hand end point of C_i precedes the right-hand end point of C_j. Thus we may associate with interleaving I a polyhedral set $K(I)$ in R^n, and, by Bayes' Theorem, $p(I|D)$, the probability of I given D is equal to

$$\frac{\int_{K(I)} p(D|x)dx}{\int_{[0,N-1]^n} p(D|x)dx}.$$

The denominator in this expression is a constant independent of I and hence can be ignored when comparing the likelihoods of interleavings. Thus, computing $p(I|D)$ involves computing a single complicated n-fold multiple integral.

4 Local Improvement Strategies

The problem of finding a most likely interleaving is very difficult. Thus we consider a number of related problems. Each of these involves defining a function $f_D(I)$, defined for every interleaving, which is simpler to deal with than $p(I|D)$ and is in some sense an approximation to $-\log p(I|D)$. The resulting problems are *NP*-hard, as is the original problem, but they are easier to deal with.

To approximate $\min_I f_D(I)$ we employ a local improvement strategy similar to the ones commonly used for solving the traveling-salesman problem. We associate with every interleaving I a unique ordering π of the clones, given by the order of occurrence of the left-hand end points in the interleaving π. Interleaving I is said to be *compatible* with the permutation π. In outline, the algorithm is as follows.

1. Let $c(\pi) = \min_I f_D(I)$, where I ranges over the interleavings compatible with π. For the objective functions we consider, $c(\pi)$ can be computed efficiently;

2. Define a *neighborhood structure* on the set of permutations of $\{1, 2, \ldots, n\}$. Such a structure may be described by a vertex-transitive graph G whose vertex set is the set

of permutations of $\{1, 2, \ldots, n\}$; adjacent vertices in this graph are considered to be neighbors.

3. Choose an initial permutation π_0 and construct a maximal path $(\pi_0, \pi_1, \ldots, \pi_t)$ in G such that, for $i = 0, 1, \ldots, t-1$, $c(\pi_{i+1}) < c(\pi_i)$;

4. Return the interleaving I of minimum cost among interleavings consistent with π_t.

To implement the approach we provide efficient algorithms for computing $c(\pi)$ (i.e., minimizing $f_D(I)$ over all interleavings compatible with π), and for searching through the neighborhood of a given permutation to determine whether a local improvement is possible.

5 Interleavings as Lattice Paths

Before defining particular objective functions, we introduce a representation of interleavings that will be used throughout the paper. We exploit the property, implicit in the Lander-Waterman model, that no clone properly contains another.

Let $M_{n,n}$ be an $n \times n$ array of cells. Define a *lattice path* as a path from the $(1,1)$-cell to the (n,n)-cell in which each edge passes from a cell to either its right neighbor or its lower neighbor, and every cell in the path is of the form (i,j) where $j \geq i-1$; i.e., the path never ventures more than one diagonal below the main diagonal. There is a natural one-to-one correspondence between lattice paths and the interleavings compatible with permutation π. Each cell visited by a lattice path corresponds to an atomic interval. Cells below the main diagonal correspond to gaps. A cell (i,j) on or above the main diagonal corresponds to an interval in which clones $\pi(i), \pi(i+1), \ldots, \pi(j)$ intersect and no other clones are present. For $j = 2, 3, \ldots, n$, a move to the right in which the path enters column j corresponds to the beginning of clone $\pi(j)$; for $i = 1, 2, \ldots, n-1$, a downward move in which the path leaves row i corresponds to the end of clone $\pi(i)$. Note that no cell of a lattice path is strictly up and to the right from another; i.e., a lattice path cannot include cells (i,j) and (i',j') such that $i' < i$ and $j' > j$.

The data D establishes that certain of the interleavings compatible with π are impossible. Suppose there is a probe A that occurs neither on clone $C_{\pi(i)}$ nor on clone $C_{\pi(j)}$ but does occur on some clone $C_{\pi(k)}$, where $\pi(i) < \pi(k) < \pi(j)$. Then $C_{\pi(i)}$ and $C_{\pi(j)}$ cannot overlap in any feasible interleaving compatible with π; for, if they did, their union would cover $C_{\pi(k)}$, and one of them would have to contain A. Thus we can exclude all lattice paths that pass strictly up and to the right from the cell $(i+1, j-1)$, since every such cell represents an atomic interval contained in $C_{\pi(i)} \cap C_{\pi(j)}$. Such a cell will be called an *excluded cell.*

Theorem 1 *A lattice path corresponds to an interleaving consistent with the data D if and only if it does not pass through any excluded cells.*

Proof:

The proof is in two parts. First assume that we have a lattice path which corresponds to an interleaving consistent with D. Suppose that the lattice path passes through an excluded

cell. An excluded cell is precisely a cell that represents an infeasible atomic interval according to the data D. We conclude that the path cannot pass through any excluded cells.

Now assume that we have a lattice path which does not pass through any excluded cells. Thus any cell through which the lattice path passes is feasible according to the data D.

Now, take any instance where $d_{ij} = 1$. Consider a cell (k, l) on the lattice path such that $k \leq i \leq l$. Suppose that there is some clone in the atomic interval corresponding to this cell which does not contain probe P_j. All the clones in this atomic interval which do not contain probe P_j must be either to the right or left of clone C_i. Otherwise cell (k, l) would have been excluded. Assume without loss of generality that they are to the left of clone C_i.

Let clone $C_{k'}$ be the rightmost clone in this atomic interval that does not contain probe P_j. Then the first cell in row $k' + 1$ of the lattice path corresponds to an atomic interval in which all the clones contain probe P_j. Otherwise, the last cell in row k' of the lattice path would correspond to an atomic interval that contained clones on both sides of clone C_i that did not contain probe P_j. Such a cell would be excluded from the lattice path. Every instance where $d_{ij} = 1$ can be satisfied by placing probe P_j in an atomic interval found this way.

■

Define the *maximum interleaving compatible with* π as the interleaving in which clones $C_{\pi(i)}$ and $C_{\pi(j)}$ intersect if and only if their intersection is not excluded; i.e., if and only if there is no positive run r whose set of clones is contained in $\{C_{\pi(i+1)}, C_{\pi(i+2)}, \ldots, C_{\pi(j-1)}\}$. The lattice path corresponding to this interleaving has on or below it all those cells that are not excluded.

Theorem 2 *For given D and π, the maximum interleaving compatible with π can be computed using time and space $O(n + \sum_i \sum_j d_{ij})$.*

Proof:

The following algorithm assumes that the clone and probe data is presented an ordered array of lists, where each element i in the array is a list of the probes in clone $C_{\pi(i)}$. The probe lists are sorted in some fixed but artibrary order. Let $\pi(0) = 0$ and $\pi(n + 1) = n + 1$. The first and last clones, C_0 and C_{n+1} are dummy clones, containing no probes. There is a dummy probe, highest in the sorting order, at the end of each probe list.

```
for i = 1 to n + 1 do
    p1 ← head(probe_list[C_π(i−1)])
    p2 ← head(probe_list[C_π(i)])
    while ((p1 ≠ dummy_probe) OR (p2 ≠ dummy_probe)) do
        if (p2 < p1)                          /* A new run has begun */
            probe_run[p2] ← i                 /* The start of the run */
            p2 ← next(probe_list[C_π(i)])
        else if (p1 < p2)                     /* A run has ended */
            add (probe_run[p1], i − 1) to maximum_run_list
            p1 ← next(probe_list[C_π(i−1)])
        else                                  /* In the middle of a positive run */
```

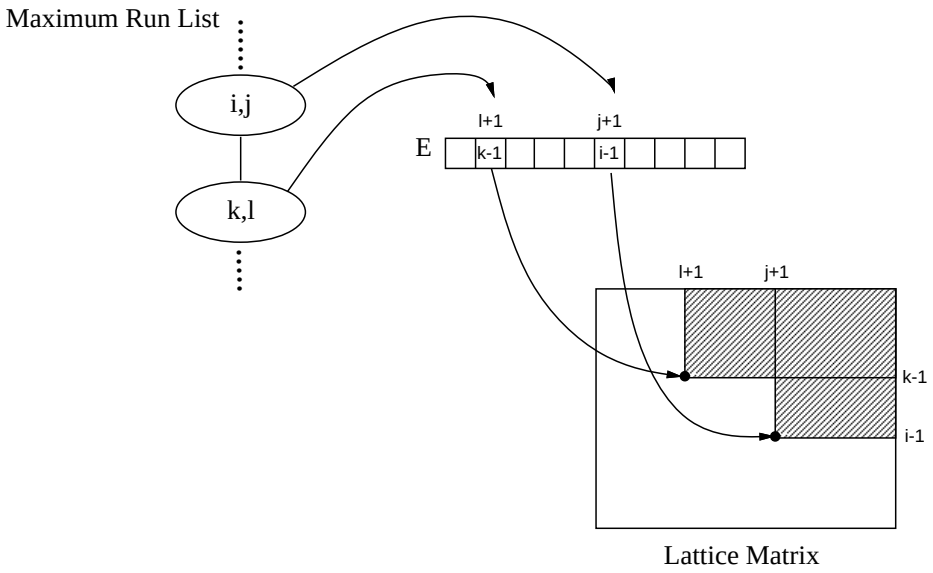


Figure 1: The relation between max_run_list, E, and the lattice path matrix.

$p1 \leftarrow \text{next}(probe\_list[C_{\pi(i-1)}])$
$p2 \leftarrow \text{next}(probe\_list[C_{\pi(i)}])$
endif
endwhile
endfor

We now calculate the excluded region. Recall that pair (i, j) denotes a maximum positive run from clone $C_{\pi(i)}$ to clone $C_{\pi(j)}$. If $\pi(i) \neq 1$ and $\pi(j) \neq n$, then the pair (i, j) marks an excluded region in the lattice path matrix, i.e. cell $(i-1, j+1)$ and all cells above and to the right of it are excluded.

We represent the excluded region in an array of size n, which we call E. Each cell in E represents a column in the lattice path matrix. The value in each cell of E is the row number which lies on the perimeter within the excluded region.

First we mark all the bottom left endpoints of local excluded regions contributed by positive runs. This takes time $O(\sum_i \sum_j d_{ij})$:

for each run (i, j) in $maximum\_run\_list$ such that $\pi(i) \neq 1$ and $\pi(j) \neq n$ **do**
  if $(E[j+1] < i-1)$ **then**
    $E[j+1] \leftarrow i-1$
  endif
endfor

Then we construct the entire excluded region in time $O(n)$:

for $i = 1$ **to** $n - 1$ **do**
  if $(E[i] > E[i+1])$ **then**
    $E[i+1] \leftarrow E[i]$
endfor

The lattice path which borders the perimeter of the excluded region represents the maximum interleaving. The path starts in the upper left cell. We construct the path so that before it enters column j it is in row $E[j]+1$. The path finally exits out of the bottom right cell. The path is constructed in time $O(n)$. Hence the entire running time of the algorithm is $O(n + \sum_i \sum_j d_{ij})$.

■

Example 1. In Figure 2 the incidence matrix, the set of positive and negative runs, the lattice path corresponding to the maximum interleaving, and the maximum interleaving are shown. The clone set is $C = \{1, \cdots, 7\}$ and probe set $P = \{A, B, C, D, E, F\}$.

	A	B	C	D	E	F	G
1	0	1	0	1	0	0	1
2	0	1	1	0	0	1	0
3	1	1	1	0	0	0	0
4	0	1	1	1	0	0	1
5	1	1	0	0	1	1	1
6	0	0	0	1	1	0	0
7	1	0	0	1	0	1	1

	Positive Runs	Negative Runs
A	3—5—7	12—4—6
B	12345	67
C	234	1—567
D	1—4—67	23—5
E	56	1234—7
F	2—5—7	1—34—6
G	1—45—7	23—6

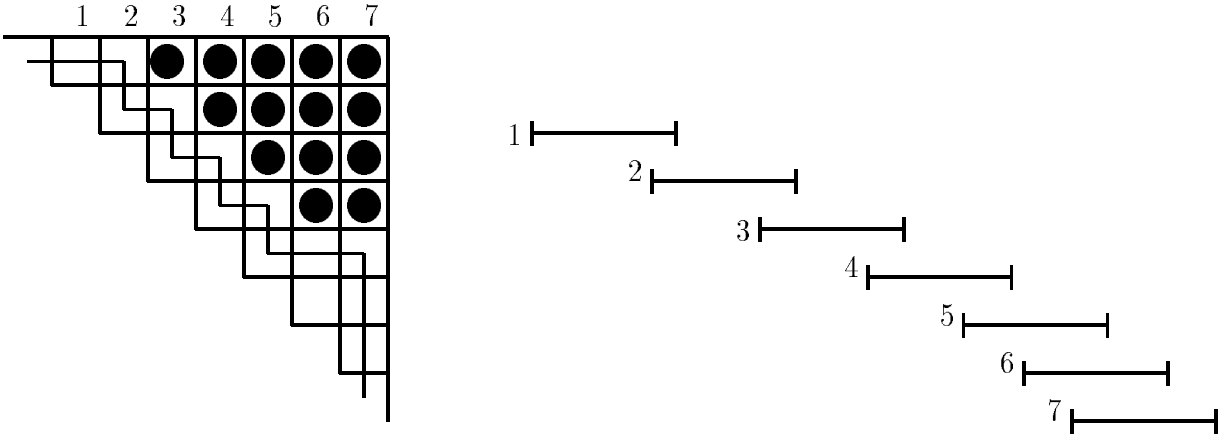


Figure 2: Lattice path corresponding to the maximum interleaving of data. The maximum interleaving is shown on the right.

Consider a fixed placement x of the clones and let I be the interleaving associated with this placement. Let π be the order in which the clones occur in x (and I). Suppose there are m probes, each occurring in accordance with a Poisson process of rate λ. Let D be

the incidence data between the clones and the probes. For the given data D, let I^* be the maximum interleaving compatible with π. We shall argue that I^*, which can be computed from π and D, is likely to resemble the true interleaving I: Any two clones C_i and C_j that overlap in x (and hence in I) will also overlap in I^*. If C_i and C_j are disjoint in x (and I) and are not consecutive in the ordering π then they will be disjoint in I^* if and only if some clone that lies between C_i and C_j in π contains a probe that occurs in neither C_i nor C_j. In this case the probability that C_i and C_j will overlap in I^* even though they are actually disjoint, goes to zero exponentially with m. On the other hand, if C_i and C_j are consecutive in the ordering then they will overlap in I^*, even if there is actually a gap between them. Thus, as m grows, I^* will eventually become identical to I, except that the gaps in I are "squeezed out" in I^*.

6 A Simple Approach Based on Hamming Distance

In this section we show that, when m the number of probes is sufficiently large, a simple approach related to the traveling salesman problem yields the true ordering of the clones with high probability. Let $A = [a_{ij}]$ be the *probe distance matrix* that is a_{ij} is the Hamming distance between row i and row j of data D:

$$a_{ij} = \sum_{k=1}^{m} (d_{ik} + d_{jk} \bmod 2).$$

Also let $A^* = [a^*_{ij}]$ be the *interval distance matrix*, that is a^*_{ij} is the *length* of the symmetric difference between clones, viewed as intervals of unit length on the real line:

$$a^*_{ij} = \max(2, |l_j - l_i| + |r_j - r_i|)$$

where l_i and r_i are the left-end and the right-end coordinates of clone c_i, respectively. We will also assume that there are no gaps in the placement, that is the underlying interleaving is connected.

Let τ be the true order of clones, that is the order in which they appear in the placement. One rough approximation to τ is the *greedy* permutation γ, which is constructed as follows: First the two clones among all pairs of clones that are closest to each other are identified and put together. (Throughout, ties are broken arbitrarily.) Then, recursively, having constructed the partial sequence of clones $c_i, \cdots, c_j$, the closest clone to either c_i or c_j is identified and attached to c_i or c_j, whichever is closer.

A slightly more accurate approximation to τ is ξ, the permutation induced by the shortest Hamiltonian cycle with respect to the Hamming metric. For this permutation, assume that we also have a fictitious empty clone added as a row of zeros to the top of the incidence data D. Then we construct a complete graph whose distance matrix is given by A (with an additional row and column corresponding to the empty clone.) The permutation ξ is obtained by solving the metric traveling salesman problem over A, and then removing the empty clone from the permutation. It turns out that ξ, herein referred to as the TSP permutation, minimizes the

total number of positive and negative runs. We will show the case for positive runs, the case for negative runs is proved similarly.

Theorem 3 *let $C = \{c_1, \cdots, c_n\}$ be the set of clones and $P = \{p_1, \cdots, p_m\}$ the set of probes. Then ξ, the permutation induced by the minimum length Hamiltonian cycle in the Hamming metric, minimizes the total number of positive runs.*

Proof: Let D be the $n \times m$ incidence matrix of clones and probes whose rows are arranged according to some permutation π. As before, we assume that the entries in the first row of D are all zero. Then for each probe p_i, if we walk from the first row through all rows and then go back to the first row, the number of transitions from 0 to 1 or from 1 to 0 will be exactly twice the number of positive runs for probe p_i. But each transition from 1 to 0 or 0 to 1 contributes 1 to the Hamming length of of the cycle induced by π. Therefore, the total number of positive runs over all probes is exactly half the length of the Hamiltonian cycle corresponding to π; in particular, the minimum length Hamiltonian cycle corresponds to a permutation that minimizes the number of positive runs. ■

Now we argue that, as the number of probes gets very large, both the greedy permutation γ and the TSP permutation ξ are likely to be identical to τ. (Here we do not distinguish between a permutation and its reverse.) First the following lemma shows that for the interval distance matrix A^* the greedy and TSP permutations are indeed equal to the true permutation.

Lemma 1 *Let $C = \{c_1, \cdots, c_n\}$ be the set of clones of unit length and A^* defined as before. Assume that there are no gaps (that is the placement is connected). Let also that γ^* and ξ^* be the greedy and the TSP permutations obtained from A^*. Then γ^*, ξ^* and τ are identical, up to reversal.*

Proof: Let l_i be the left-end coordinate of clone c_i. Then, $a^*_{ij} = 2\max(1, |l_i - l_j|)$. Therefore, the problem of finding the shortest Hamiltonian cycle based on A^* is equivalent to the special case Euclidean traveling salesman problem where all cities are on a line (that is the one-dimensional traveling salesman problem; see Figure 3.) But in this case the problem is trivial: the optimal tour is the one that starts from one of the two extreme cities and goes through cities as they appear on the line and then back to the first city. Clearly the greedy algorithm can find this tour from A^*. Therefore, up to reversal $\gamma^* = \xi^* = \tau$. ■

We should remark that in order for the greedy and TSP permutations to be identical to the true permutation, we do not need to have A^*. Rather, to get the true permutation, we can apply the greedy or the traveling salesman algorithm to any matrix $A = [a_{ij}]$ with the property that for all indices i, j, k, l, $a_{ij} > a_{kl}$ if and only if $a^*_{ij} > a^*_{kl}$.

Theorem 4 *Let $C = \{c_1, \cdots, c_n\}$ be a fixed set of clones represented by a set of unit length intervals whose underlying interleaving has no gaps, and let A^* be their interval distance matrix. Suppose P is a set of m probes distributed according to a Poisson process of rate λ, and $A^{(m)}$ the corresponding probe distance matrix. If $a^*_{ij} < a^*_{kl}$, then as $m \to \infty$,*

$$\Pr\left[a^{(m)}_{ij} > a^{(m)}_{kl}\right] \to 0$$

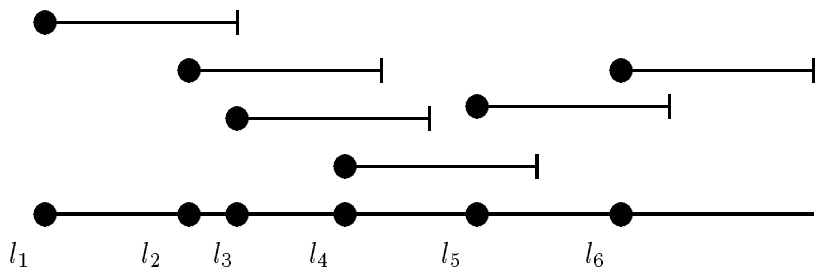


Figure 3: Mapping the interval distance to 1-dimensional Euclidean distance.

Proof: For a pair of clones c_i and c_j the probability that a probe p contributes to their hamming distance is given by

$$p_{ij} = 2e^{-\lambda}(1 - e^{-\lambda(1-x_{ij})})$$

where x_{ij} is the length of the intersection of intervals representing clones c_i and c_j. Thus the events that probes $p_1, p_2, \cdots$ contribute 1 unit to the Hamming distance between clones c_i and c_j may be viewed as a sequence of independent Bernoulli trials of probability p_{ij}. Therefore, as $m \to \infty$, by the law of large numbers, for any fixed small positive real number ϵ

$$\Pr\left[\left|\frac{a_{ij}^{(m)}}{m} - p_{ij}\right| > \epsilon\right] \to 0$$

But, since $a_{ij}^* = 2 - 2x_{ij}$, $a_{ij}^* > a_{kl}^*$ if and only if $p_{ij} > p_{kl}$, and therefore, the theorem is proved. ■

Hence, by the remark preceding Theorem 4, as m grows the probability that γ and ξ are equal to each other and to the true permutation tends to 1.

7 A Combinatorial Objective Function

Our problem is to maximize $p(I|D)$. Since $p(I|D)$ appears hard to compute even when I is given we do not expect to solve this maximization problem exactly. Instead, we introduce a different objective function $f_D(I)$ that is easier to handle, and argue that the interleaving that minimizes $f_D(I)$ should come close to maximizing $p(I|D)$.

By Bayes' Theorem, $P(I|D) = \frac{P(D|I)P(I)}{P(D)}$. Since D is fixed, maximizing $P(I|D)$ is equivalent to maximizing $P(D|I)P(I)$. When the number of clones is at all large $P(D|I)$ varies much more among interleavings than does $P(I)$. Thus, as an approximation, we neglect the *a priori* probability of an interleaving and seek to maximize $P(D|I)$ rather than $P(I|D)$.

Fix an interleaving I. We shall express the event D as a conjunction of nearly independent events associated with distinct runs. For any negative run (r, A), the event $GOOD(r, A)$ occurs if no clone in the run r contains an occurrence of A. For any positive run (r, A),

the event $GOOD(r, A)$ occurs if, for every clone C in the run, probe A occurs in some atomic interval which is contained in C but is not contained in any of the clones in the neighboring negative runs. Then $D = \cap_{(r,A)} GOOD(r, A)$ and we approximate $P(D|I)$ by $\prod_{(r,A)} p(GOOD(r, A))|I)$. We seek to maximize $\prod_{(r,A)} p(GOOD(r, A))|I)$.

We now introduce a simple approximation to $p(GOOD(r, A))|I)$. Define a *negative elementary event* as an event of the form "clone C_i does not contain probe P_j" (i.e., $d_{ij} = 0$) and a *positive elementary event* with respect to I as an event of the form "atomic interval U contains probe P_j,"; such an event implies $d_{ij} = 1$ for every clone C_i containing atomic interval U. For a negative run (r, A) define $c^-(r, A)$ as the minimum number of negative elementary events needed to imply $GOOD(r, A)$. Then $c^-(r, A)$ is an approximation to the length of run (r, A), and thus we approximate $p(GOOD(r, A)|I)$, the conditional probability that A does not occur in any clone of r, by $e^{-\phi c^-(r,A)}$ where ϕ is a constant closely approximating the probe hybridization rate, λ. Similarly, for a positive run (r, A), let $c^+(r, A)$ be the minimum number of positive elementary events with respect to I needed to imply $GOOD(r, A)$, given the interleaving I. Then we approximate $p(GOOD(r, A)|I)$ somewhat crudely by $e^{-\theta c^+(r,A)}$, where $e^{-\theta}$ is an approximation to the probability that an atomic interval contains probe A. Thus our approximation to $p(D|I)$ is $e^{-(\phi \sum c^-(r,A) + \theta \sum c^+(r,A))}$, where the first sum is over negative runs and the second sum is over positive runs. Taking logarithms and changing sign, we define $f_D(I)$ as $\phi \sum c^-(r, A) + \theta \sum c^+(r, A)$ and take it as our goal to find an interleaving I that minimizes $f_D(I)$.

Theorem 5 *Let I^* be the maximum interleaving with respect to π. Then there is a set of elementary events S^* relative to I^* such that (S^*, I^*) implies D, and, for any interleaving I compatible with π and D, and any set S of elementary events relative to I such that (S, I) implies D, S contains at least as many positive elementary events as S^* does and S contains at least as many negative elementary events as S^* does. In other words, $\sum c^+$ and $\sum c^-$ are minimized in I^*.*

Proof:

Define S_I^- and S_I^+ to be minimum sets of negative and positive elementary events for an interleaving I that imply the data D. Let I be any interleaving compatible with π and D. We want to prove that $|S_{I^*}^+| \leq |S_I^+|$ and $|S_{I^*}^-| \leq |S_I^-|$ for all interleavings I compatible with π and D.

S^-: Every negative elementary event in I^* is covered by S_I^- in one of two ways: Either it is an element of S_I^-, or it is covered by two elements in S_I^-. Since containment is not permitted, exactly two events are required to cover a third event. Any clone which is covered by two clones in I must also be covered by those clones in I^*, since I^* is maximum. Likewise any event in I which is covered by two elements in S_I^- must be covered in I^* as well.

S^+: For every positive elementary event in I "atomic interval U contains probe j," there is a corresponding positive elementary event in I^* "atomic interval V contains probe j," where the clones in U are a subset of the clones in V.

To see that such a V exists, first consider any atomic interval U^* in I^* which contains all the clones of U. There must be such a U^* because I^* is maximum.

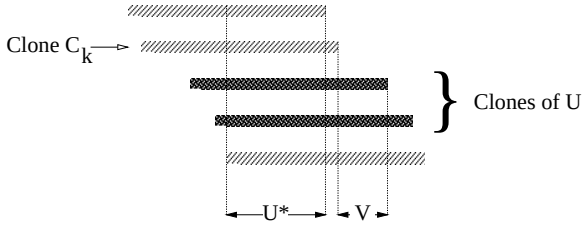


Figure 4: C_k, V, U^*, and the clones of U in I^*

Suppose that U^* cannot contain probe j because it includes a clone which does not contain j. The clones in U^* which do not contain probe j must be on one side or the other of the clones of U. Otherwise the cell representing U^* would have been excluded when calculating the lattice path.

Assume without loss of generality that the clones in U^* which do not contain probe j are to the left of the clones of U. Let C_k be the clone closest to the clones in U that does not contain j.

Moving from left to right from U^*, we claim that V is the first atomic interval which does not contain clone C_k. V includes all the clones of U and no clones which do not contain probe j. If V did include a clone which did not contain probe j, it would have to be a clone to the right of U. Then the atomic interval to the left of V would have been forbidden, for it would have included clones on either side of U which did not contain probe j.

Since every positive elementary event in I^* can be covered by one positive elementary event in I^*, $S^+_{I^*}$ is no larger than S^+_I.

Since f exists for all I consistent with π and D, $S^+_{I^*}$ is no larger than S^+_I.

■

Corollary 1 *$f_D(I^*)$ is the minimum, over all I compatible with π of $f_D(I)$.*

Proof: c^+ and c^- are both minimized in I^*. ■

Theorem 6 *The problem of finding an interleaving that minimizes $f_D(I)$ over all I is NP-hard.*

Proof:

The proof is by reduction from the special case of the Hamiltonian Path problem where each vertex has degree at least 3. The vertices map to clones, and the edges map to probes. We create the edge incidence matrix D from $G = (V, E)$. D is an m by n matrix, where $m = |E|$ and $n = |V|$. $d_{ij} = 1$ and $d_{ik} = 1$ for the ith edge $(j, k) \in E$. All other entries in row i are 0.

This is a special case in which each probe is incident to a unique set of two clones. Thus each clone can have a non-empty overlap with at most two other clones. The sum $\sum_{I^*} c^+$ is minimized over all interleavings when every clone except the first and last has a non-empty

overlap with two other clones. This is precisely the case in which there is one connected component of clones, and thus the graph corresponding to the interleaving has a Hamiltonian Path. We conclude that $\sum_{I^*} c^+ \geq (n-1)+2(m-n+1)$, and there is one connected component of clones if and only if equality holds.

We assign a zero weight to the $\sum c^-$ term of $f_D(I)$ so that the graph corresponding to an interleaving has a Hamiltonian Path if and only if $f_D(I)$ is minimized.

■

Theorem 7 *Given interleaving I and data D, one can compute $f_D(I)$ in time $O(n+\sum_{ij} d_{ij})$.*

The algorithm works separately on each run with respect to π. For each negative run (r, A) it places in S the minimum number of negative elementary events needed to imply that no clone in r contains A. For each positive run, it places in S the minimum number of positive elementary events consistent with D needed to imply that each clone in r contains A. In both the positive and negative cases, dynamic programming is used to construct these sets. We omit the details.

Let $c(\pi)$ be the minimum, over all interleavings I compatible with π, of $f_D(I)$. Then $c(\pi) = f_D(I^*)$. Combining Theorems 2 and 7 we find that $c(\pi)$ can be computed in time $O(n + \sum_{ij} d_{ij})$.[1]

8 Computing $\min_\pi c(\pi)$

We continue to work with the objective function $f_D(I)$ and the associated function $c(\pi)$. Our algorithmic approach to minimizing $c(\pi)$ resembles the 2-opt, 3-opt and Lin-Kernighan local improvement algorithms that are widely used to solve the symmetric traveling-salesman problem ([LK73]). Our algorithms require a neighborhood structure on the set of permutations of $\{1, 2, \ldots, n\}$. Possible choices are:

The 2-opt neighborhood structure Permutations π and σ are neighbors if and only if σ can be obtained from π by subdividing π into three parts and reversing the middle part; i.e., if and only if π can be written as the concatenation of α, β and γ, and σ as the concatenation of α, β^R and γ, where β^R is the reversal of β.

the 3-opt neighborhood structure Permutations π and σ are neighbors if and only if σ can be obtained from π by subdividing π into four parts, possibly reversing one or both of the middle parts, and possibly interchanging the two middle parts.

The local improvement procedure starts with a permutation π_0; at a general step, given a permutation π_i, it searches for a permutation π_{i+1} such that π_i and π_{i+1} are neighbors and $c(\pi_{i+1}) < c(\pi_i)$. The computation terminates at a local minimum when no such neighbor exists.

[1] We are grateful to Eugene Lawler for simplifying our original algorithm for computing $c^+(\pi)$ and improving the space requirement.

The main challenge in implementing our local improvement procedures is to search efficiently through the neighborhood of the present permutation, implicitly or explicitly computing the cost of each neighbor. A key observation is that $c(\pi)$ is a sum of independent terms corresponding to the runs associated with π. Thus, in computing $c(\sigma)$ knowing $c(\pi)$, it is sufficient to add in a term for each run created, and subtract out a term for each run destroyed, in moving from σ to π.

9 An Objective Function Based on Pairwise Overlap Probabilities

Let C_i and C_j be two clones. Let a_0 be the number of probes that occur on neither C_i nor C_j, a_1, the number of probes on exactly one of the two clones, and a_2, the number of probes that occur on both C_i and C_j. Within the Lander-Waterman stochastic model, let p_{ij} be the conditional probability that C_i and C_j overlap, given this data. Let O be the event that clones C_i and C_j overlap. A calculation based on Bayes' Theorem gives

$$p_{ij} = \frac{p(O)p(D|O)}{p(O)p(D|O) + p(\overline{O})p(D|\overline{O})}$$

where

$$p(\overline{O}) = \left(\frac{N-2}{N-1}\right)^2, \qquad p(O) = 1 - p(\overline{O}),$$

$$p(D|\overline{O}) = e^{-\lambda(2a_0+a_1)}(1-e^{-\lambda})^{a_1+2a_2},$$

$$p(D|O) = \int_0^1 e^{-\lambda(2-x)a_0} e^{-\lambda a_1}(1-e^{-\lambda(1-x)})^{a_1}[(1-e^{-\lambda x}) + e^{-\lambda x}(1-e^{-\lambda(1-x)})^2]^{a_2} dx$$

Thus p_{ij} can be computed easily with the help of numerical integration. Let $q_{ij} = 1 - p_{ij}$. We can approximate $p(I|D)$ by

$$f'_D(I) = \prod_{E(I)} p_{ij} \prod_{\overline{E(I)}} q_{ij}$$

where $E(I)$ denotes the set of pairs of clones that overlap in the interleaving I and $\overline{E(I)}$ denotes the set of pairs of clones that do not overlap in the interleaving I.

This approximation to $p(I|D)$ is based on the faulty assumption that the events of the form 'C_i and C_j overlap' are independent. This independence assumption can lead to nonsensical results. For example, consider the case where probe P_1 is incident with C_1 but not C_2 or C_3, P_2 is incident with C_2 but not C_1 or C_3, probe P_3 is incident with C_3 but not C_1 or C_2, and each of the $m-3$ remaining probes is incident with C_1, C_2 and C_3. When m is sufficiently large p_{12}, p_{13} and p_{23} will all be close to 1, since each of these pairs of clones has $m-3$ probes in common. Nevertheless, it is clear that all three pairs cannot overlap; for, if they did, then one of the clones would be covered by the union of the other two, and thus every

probe incident with that clone would be incident with one of the other two, contradicting the given data.

Despite its theoretical drawbacks this approximation to $p(I|D)$ is widely used in physical mapping ([BSPGCW90, CdJBSW89, TOTdJTZCM92]) and has the practical advantage that diverse kinds of fingerprint data (and even noisy data) can be combined in order to estimate the p_{ij}; once the p_{ij} are determined the original data is no longer taken into consideration.

We consider the problem of finding an interleaving to maximize $f'_D(I)$. Letting $w_{ij} = \ln(\frac{p_{ij}}{q_{ij}})$, we obtain the following equivalent problem: find an interleaving I to maximize $\sum_{\{i,j\}\in E(I)} w_{ij}$.

Theorem 8 *The problem of finding an interleaving to maximize $\sum_{\{i,j\}\in E(I)} w_{ij}$ is NP-hard.*

Proof: The proof is by reduction from the Hamiltonian path problem in bipartite graphs. Let $G_1 = (V, E)$ be a bipartite graph, construct a weighted complete graph $G = (V, w)$, with $w_{ij} = 1$ if i, j is an edge in the bipartite graph and $w_{ij} = 0$, otherwise. Then the maximum weight proper interval subgraph in G corresponds to a union of paths in G_1. Thus, G_1 is Hamiltonian if and only if the maximum interleaving is connected, that is the union of paths is a singleton. ■

Theorem 9 *For a given permutation π the problem of finding an interleaving compatible with π to maximize $\sum_{\{i,j\}\in E(I)} w_{ij}$ is solvable in time $O(n^2)$.*

Proof: Because of the correspondence between lattice paths and interleavings compatible with π, we can restate the problem as follows: given a $n \times n$ upper triangular matrix (w_{ij} with zeros on the main diagonal, find a lattice path P which maximizes $\sum_{L(P)} w_{ij}$, where $L(P)$ is the set of cells on or below P. Algorithm I, which is based on shortest-path techniques, solves the problem in time $O(n^2)$.

Algorithm I

- Construct a $n \times n$ upper triangular matrix (v_{ij}) by computing partial sums along upward diagonals in the matrix (w_{ij}); i.e., $v_{ij} = \sum_{h=0}^{\lfloor \frac{j-i}{2} \rfloor} w_{i+h,j-h}$.
- Using a standard algorithm for computing longest paths in acyclic digraphs, find a lattice path of maximum total weight, where the weight of cell (i, j) is v_{ij}. Set P equal to this lattice path.

■

Example 2. Figure 5 demonstrates Algorithm I. The table on the left contains the v_{ij}. Arrows in that table indicate the direction that the partial sums should be accumulated. The table on the right contains w_{ij} and the shortest lattice path obtained by applying Algorithm I.

The following algorithm, which is somewhat more efficient than Algorithm I, computes the optimal value of the objective function $\sum_{L(P)} w_{ij}$ without determining the optimal lattice

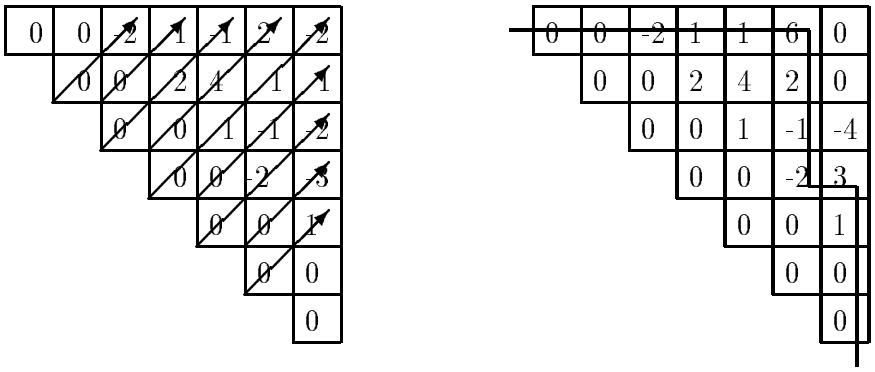


Figure 5: An example of Algorithm I.

path P. It proceeds by a sequence of transformations of the matrix, each of which preserves the optimal value but may change the optimal path.

Algorithm II

Border the matrix with a zeroth row whose entries are all zero;

for $i = n$ **down to** 1 **do**
  for $j = n$ **down to** i+1 **do**
    if $w_{ij} > 0$ **then**
      $w_{i,j-1} \leftarrow w_{i,j-1} + w_{ij}$
    else
      $w_{i-1,j} \leftarrow w_{i-1,j} + w_{ij}$
    endif
    $w_{ij} \leftarrow 0$
  endfor
endfor
return $\sum_i w_{ii}$

Let $c(\pi)$ be the maximum, over all interleavings compatible with π, of $\sum_{(i,j)\in E(I)} w_{ij}$. Algorithm I or Algorithm II can be used to compute $c(\pi)$. A local improvement method based on the 2-opt, 3-opt or Lin–Kernighan neighborhood structure can be used to compute a good approximation to $\max_\pi c(\pi)$, and to determine the interleaving that yields this maximum value.

10 Some Computational Results

In this section we present some computational results based on our experiments with the $c^+ + c^-$ objective function. For comparison we also include one example from the greedy and the traveling salesman approaches as discussed in section 6.

λ	c	$m = 20$			$m = 30$			$m = 50$			$m = 70$		
		c_1	c_2	c_3	c_1	c_2	c_3	c_1	c_2	c_3	c_1	c_2	c_3
0.29	2.5	739	630	672*	1224	994	1040*	2184	1756	1793*	3127	2506	2552*
	5.0	528	434	419	710	625	631*	1280	1066	1066	1799	1543	1543
	10.0	280	234	234	393	342	342	705	597	597	920	823	823
0.69	2.5	774	623	713*	1214	1014	1077*	2130	1778	1828*	3223	2560	2587*
	5.0	564	469	432	860	672	669	1226	1131	1133*	1689	1671	1627
	10.0	338	254	254	516	394	394	735	712	674	1022	984	935

Figure 6: $c_1 = c^+ + c^-$(Greedy), $c_2 = c^+ + c^-$(2-opt), and $c_3 = c^+ + c^-$(True).

The experiments were conducted on simulated, rather than real data. Using Lander–Waterman model, we generated placements for n unit-length clones whose left–ends were uniformly and independently distributed over the interval $[0 \cdots N-1]$ (thus, the original chromosome is represented by the interval $[0 \cdots N]$.) To assign m probes to the n clones according to the Poisson distribution of rate λ, we calculated the lengths of the $2n-1$ atomic intervals. Then for each atomic interval A of length l , and each probe P, we assigned P to A–and to all clones covering A–with probability $1-e^{-\lambda l}$. Labeling clones from left to right by 1,2, ..., n, the true permutation of clones was always $\tau := [12 \cdots n]$ or the reverse permutation $n \cdots 21$. (As usual, we do not distinguish between a permutation and its reverse). This labeling makes it convenient to compare the true permutation to the permutations generated by our algorithms.

In each experiment we started with the greedy permutation γ, which was generated by the greedy algorithm. We then used γ as the starting point of the 2-opt local search algorithm with the $c^+ + c^-$ objective functions.

We generated data for two values of λ, namely $-\ln(0.25) \approx 0.29$ and $-\ln(0.5) \approx 0.69$ corresponding to probabilities 25% and 50%, respectively, that a given probe appears in a chromosome segment of unit length. For each λ we generated data for two values of coverage, $c = 5$ and $c = 10$. Throughout we fixed the number of clones at $n = 100$, hence, altogether we experimented on four different placements. For each placement, we varied the number of probes over $\{10, 20, 30, 50, 70\}$. The permutations σ generated by the 2-opt heuristic are shown in Figures 8 through 11. In each graph the the x-axis corresponds to i and the y-axis corresponds to σ_i. Since the true permutation $\tau = 12 \cdots n$ (or its reverse), the perfect solution would be either a 45 degree line through the origin or a -45 degree line from point $(1, n)$ to point $(n, 1)$. Therefore, in these graphs long stretches of ± 45 degree line segments indicate the success of the 2-opt heuristic in zeroing in on the true permutation, and scattered short line segments mean a less successful attempt.

Overall, $c^+ + c^-$ seems to be a good objective function, except for small values of m or low coverage. In the table in Figure 6 the $c^+ + c^-$ value of the permutation generated by the 2-opt heuristic (c_2) is compared with the $c^+ + c^-$ values of the greedy (c_1) and the true permutations (c_3). Some entries under c_3 columns are marked by "*". This indicates that in those cases, the $c^+ + c^-$ value of 2-opt permutation is smaller than that of the true permutation.

In Figure 7 the results of greedy and the traveling salesman algorithms are compared with

the $c^+ + c^-$ 2-opt local optimum. For the traveling salesman problem, a local search based on the Lin-Kernighan heuristic was used. In this single experiment we used 200 clones, 40 probes, rate $-\ln(0.50)$, and coverage 5. Clearly, for this experiment $c^+ + c^-$ is a superior objective function. However, we have also observed that for large m and higher coverage, the TSP permutation, and sometimes even the greedy permutation are already very close to the true permutation. On the other hand for small values of m and low coverage all three permutations are fairly poor.

Our experiments seem to hint at the following phenomenon. For small values of m, there is simply not enough information in the data to reveal the placement of clones, and therefore, all algorithms are expected to perform poorly. For very large m, the argument of section 6 shows that even the most simple–minded method such as the greedy algorithm is likely to find the true permutation. However, it seems that there is a range of values for m, depending on the underlying placement, where the result of the $c^+ + c^-$ objective is superior to those of the greedy or the traveling salesman heuristics.

11 Future Work

Our eventual goal is to provide researchers in molecular biology with powerful algorithmic tools for the physical mapping of DNA molecules. The present paper introduces an approach based on minimizing an objective function of the form $\phi \sum c^-(r, A) + \theta \sum c^+(r, A)$. Our initial experiments using synthetic data indicate that the approach holds promise. However, to exploit the approach fully will require the efficient implementation of local search methods more powerful than 2-opt. It will also be necessary to extend the methods to work in the presence of errors in the incidence data between probes and clones. The final test will be the successful application of the approach to real data from the genomes of humans and other species.

We also hope to understand our methods better from a theoretical point of view. For example, in light of the NP-hardness of minimizing $\phi \sum c^-(r, A) + \theta \sum c^+(r, A)$, we would like to know whether there is a polynomial-time algorithm to approximate the optimal value of this objective function to within a constant factor.

With respect to the the Lander-Waterman stochastic model introduced in Section 2, we would like to prove a threshold property of the following form: for given values of n, the number of clones, c, the coverage, and λ, the Poisson rate of occurrence for each probe , there is a critical value $m(n, c, \lambda)$ such that, if m, the number of probes, is below this critical value then the data will almost surely not contain enough information to determine the true interleaving, whereas, if m is significantly above the critical value, then it will be possible to identify the true interleaving correctly with high probability. We would also like to compare various mapping algorithms according to the number of probes they require (as a function of n, c and λ) in order to determine the true interleaving with high probability.

We hope to explore further the approach introduced in Section 9, based on minimizing an objective function of the form $\sum_{\{i,j\} \in E(I)} w_{ij}$. Such an objective function has the advantage of flexibility, since a variety of fingerprinting technologies can be used to derive pairwise overlap

probabilities.

We also intend to explore the algorithmic problems associated with physical mapping using Sequence Tagged Sites. This mapping technique tends to produce unique probes; i.e., probes that occur at only one location along the DNA molecule. Some initial computational experiments indicate that it may be relatively easy to construct good physical maps from such fingerprint data, even in the presence of a rather high rate of error in the clone-probe incidence data.

Finally, we recognize that the role of combinatorial optimization in physical mapping is not to construct correct physical maps automatically, but to aid biologists by suggesting possible solutions that must be confirmed or disconfirmed by further experiments. Thus, in the long run, it will be necessary to provide an interactive computing environment that enables a close interaction between the biologist and his computational tools.

Acknowledgements

We would like to thank Luciano Margara and Bruno Codenotti for making their Lin-Kernighan code available to us.

References

[BJLTY91] Avrim Blum, Tao Jiang, Ming Li, John Tromp, and Mihalis Yannakakis. Linear approximation of shortest superstrings. In *Proceedings of the Twenty-Third Annual ACM Symposium on Theory of Computing*, pages 328–336, New Orleans, LA, May 1991. ACM Press.

[BSPGCW90] E. Branscomb, T. Slezak, R. Pae, D. Galas, A. V. Carrano, and M. Waterman. Optimizing restriction fragment fingerprinting methods for ordering large genomic libraries. *Genomics*, 8(2):351–366, October 1990.

[CdJBSW89] A. V. Carrano, P. J. de Jong, E. Branscomb, T. Slezak, and B. Watkins. Constructing chromosome- and region-specific cosmid maps of the human genome. *Genome*, 31(2):1059–1065, 1989.

[CNHZL90] A. G. Craig, D. Nizetic, J. D. Hoheisel, G. Zehetner, and H. Lehrach. Ordering of cosmid clones covering the Herpes simples virus type-I (HSV-I) genome — a test case for fingerprinting by hybridisation. *Nucleic Acids Research*, 18(9):2653–2660, May 11 1990.

[LK73] S. Lin and B. W. Kernighan. An effective heuristic algorithm for the traveling-salesman problem. *Operations Research*, 21(2), March-April 1973.

[LW88] Eric S. Lander and Michael S. Waterman. Genomic mapping by fingerprinting random clones: A mathematical analysis. *Genomics*, 2(3):231–239, April 1988.

[TOTdJTZCM92] Katherin Tynan, Anne Olsen, Barbara Trask, Pieter de Jong, John Thompson, Wofgang Zimmermann, Anthony Carrano, and Harvey Mohrenweiser. Assembly and analysis of cosmid contigs in the CEA-gene family region of human chromosome 19. *Nucleic Acids Research*, 20(7):1629–1636, April 11 1992.

[Tur89] J. S. Turner. Approximation algorithms for the shortest common superstring problem. *Information and Computation*, 83(1):1–20, October 1989.

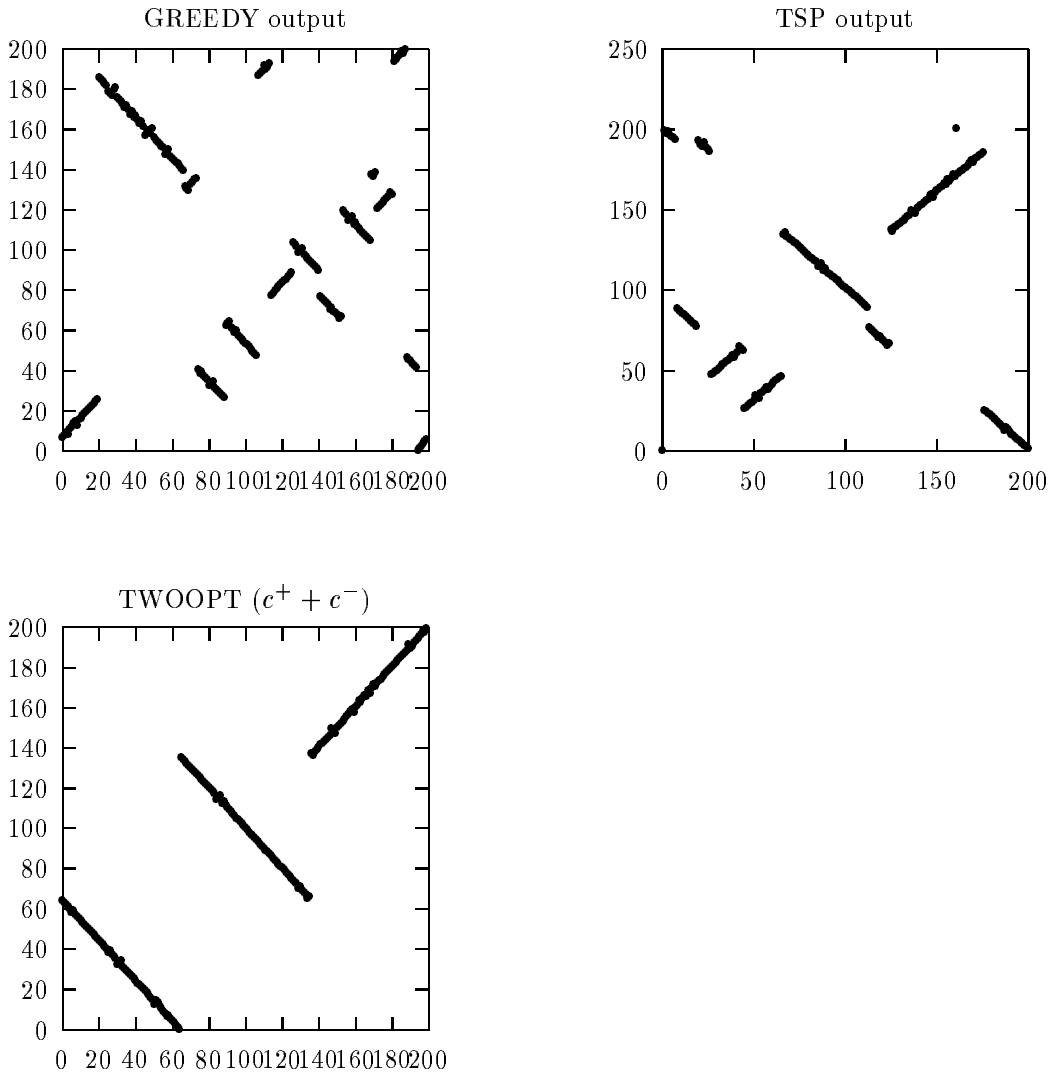
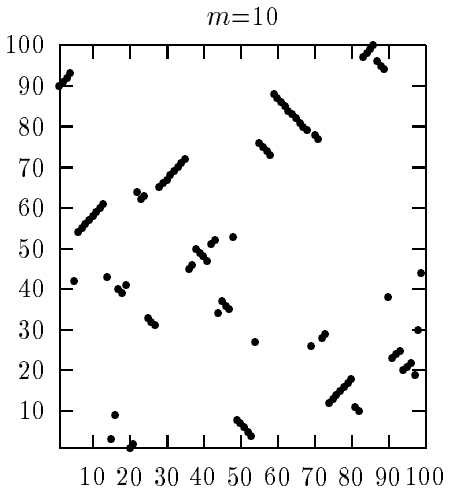
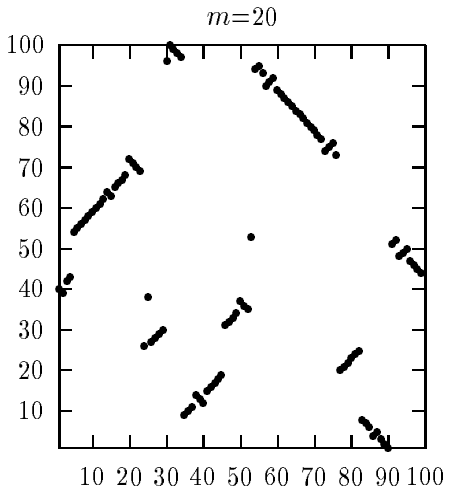
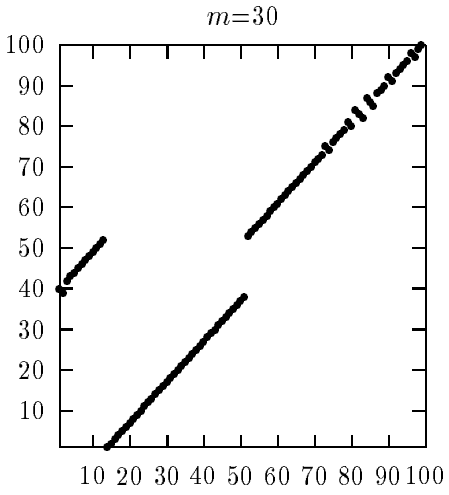


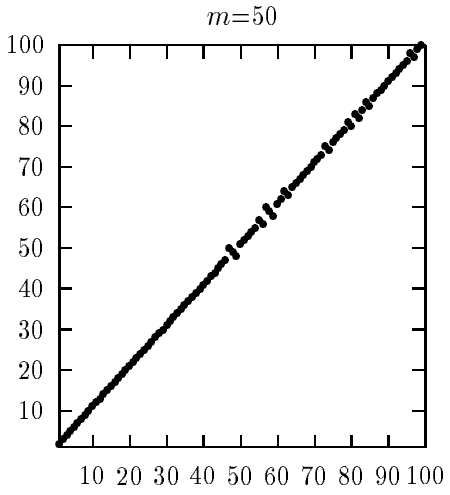
Figure 7: $\lambda = 0.69$, $c = 5.0$, $n = 200$, $m = 40$

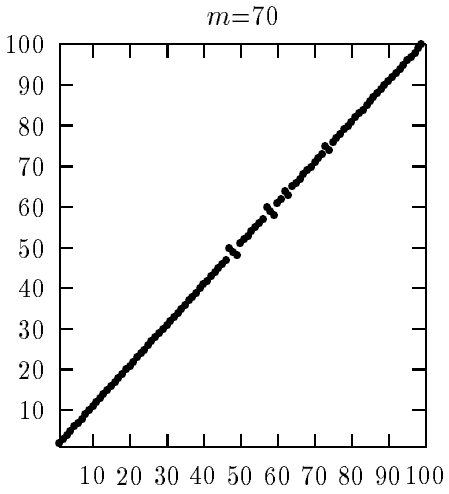
$\lambda = 0.29$, $c = 5.0$, $n = 100$



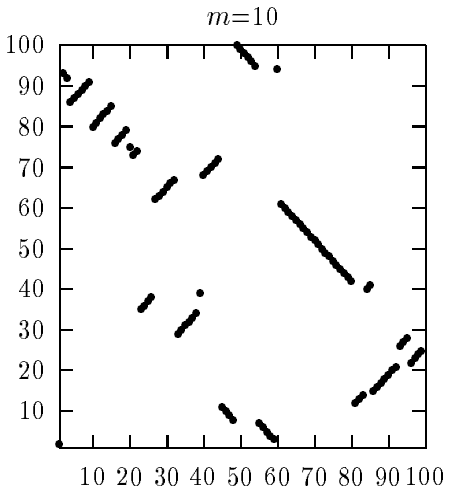


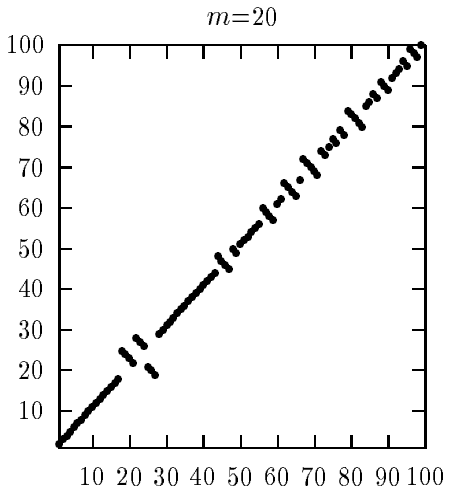


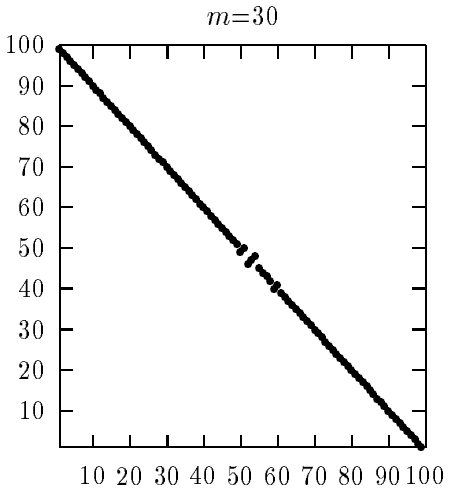


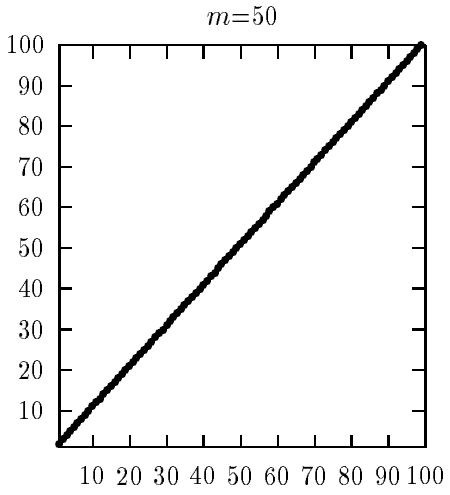


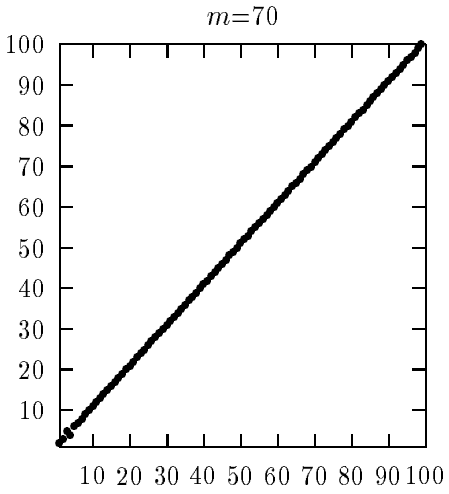
$\lambda = 0.29$, $c = 10.0$, $n = 100$



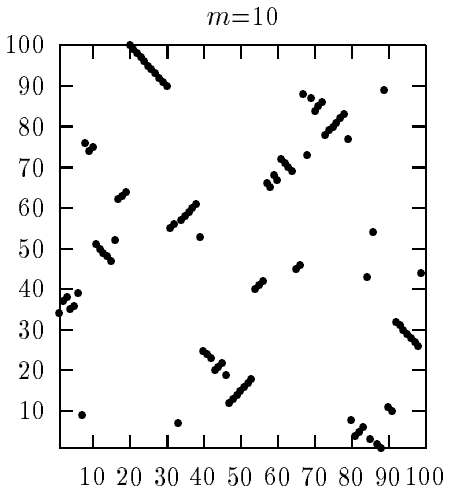


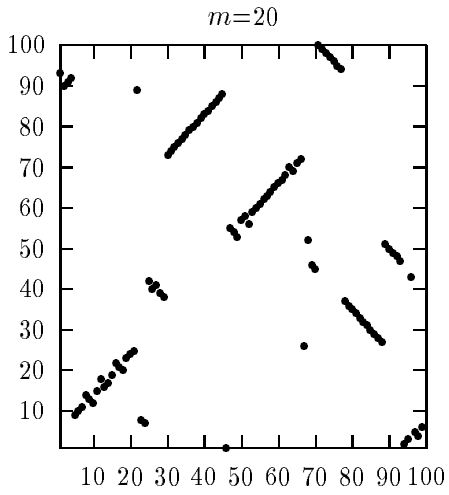


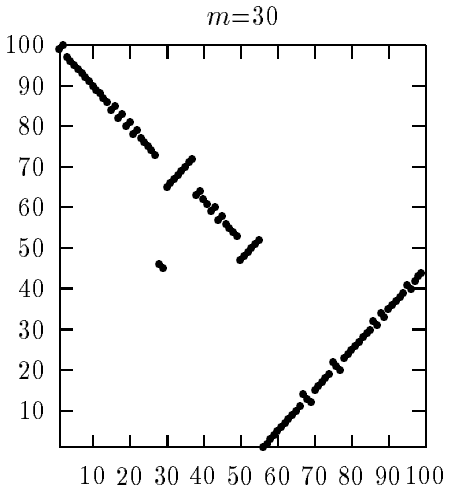


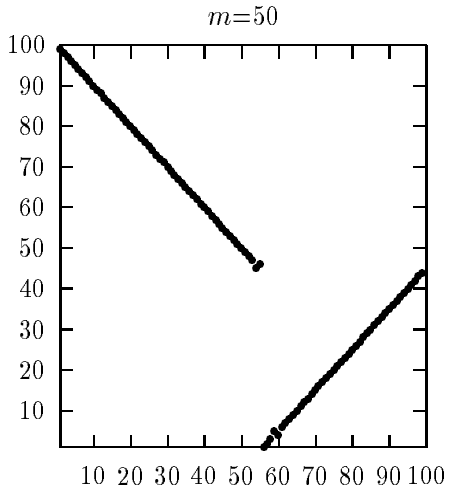


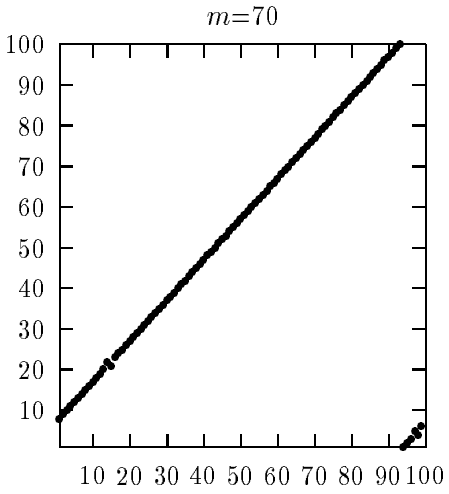
$\lambda = 0.69$, $c = 5.0$, $n = 100$











$\lambda = 0.69$, $c = 10.0$, $n = 100$