INTERNATIONAL COMPUTER SCIENCE INSTITUTE

1947 Center St. • Suite 600 • Berkeley, California 94704-1198 • (510) 643-9153 • FAX (510) 643-7684

Physical Mapping of Chromosomes Using Unique Probes

Farid Alizadeh [*] Richard M. Karp [* † ‡]

Deborah K. Weisser [†] Geoffrey Zweig [† ‡]

TR-95-015

April 1995

Abstract

The goal of physical mapping of the genome is to reconstruct a strand of DNA given a collection of overlapping fragments, or clones, from the strand. We present several algorithms to infer how the clones overlap, given data about each clone. We focus on data used to map human chromosomes 21 and Y, in which relatively short substrings, or probes, are extracted from the ends of clones. The substrings are long enough to be unique with high probability. The data we are given is an incidence matrix of clones and probes.

In the absence of error, the correct placement can be found easily using a PQ-tree. The data is never free from error, however, and algorithms are differentiated by their performance in the presence of errors. We approach errors from two angles: by detecting and removing them, and by using algorithms which are robust in the presence of errors.

We have also developed a strategy to recover noiseless data through an interactive process which detects anomalies in the data and retests questionable entries in the incidence matrix of clones and probes.

We evaluate the effectiveness of our algorithms empirically, using simulated data as well as real data from human chromosome 21.

[*] International Computer Science Institute. Research supported in part by NSF grant no. CDA-9211106.

[†] Computer Science Division, University of California, Berkeley. Research supported in part by NSF grant no. CCR-9005448.

[‡] Research supported in part by DOE Grant DE-FG03-94ER61913

1 Introduction

1.1 The Physical Mapping Problem

In this paper we present several combinatorial algorithms for reconstructing a DNA strand given a collection of overlapping fragments of the strand. A human chromosome, which is a DNA molecule of about 10^8 base pairs, is too long to be studied in its entirety and must be broken into fragments or *clones*. Depending on the cloning technology used, the sizes of the clones may be as small as $3,000$ base pairs or as large as $2,000,000$ base pairs. Information is gathered from the individual clones, and then the DNA is reconstructed by mathematically determining the positions of the clones.

The goal of *physical mapping* is to infer how the clones overlap to form the DNA molecule, given data about each clone. The present paper focuses on the Sequence Tagged Site (STS) mapping strategy, which is widely used for physical mapping within the Human Genome Project and other related molecular biology projects [PSM+91], [MCG+93]. In particular the recent mapping of human chromosome 21 [CRG+92] and human chromosome Y [VFH+92, FVHP92] use this strategy. In the STS approach relatively short substrings called *probes* are extracted from the DNA strand itself, often from the endpoints of clones. Each probe is sufficiently long that it is highly unlikely to occur a second time on the DNA strand; thus it identifies a unique site along the DNA strand. A probe is said to occur on a clone if it matches a substring in that clone. This occurrence can be tested either by hybridization or PCR experiments. We present algorithms to determine probe ordering, given data for each clone indicating which probes occur on it.

In the absence of errors, the correct orderings can be found very easily using the PQ-tree data structure [BL76] to generate the set of all arrangements of probes consistent with the data. The data is never free from error however, and algorithms are differentiated by their performance in the presence of errors.

The most common type of error is a false negative, where an incidence between a probe and a clone occurs but is not observed to occur. False positives also occur. In addition, in some cloning technologies the data is subject to error due to clone abnormalities, such as *chimeric* clones, which consist of two distinct segments of the DNA strand joined together. Deletions, insertions, and inversions of clone segments also occur.

In this paper we present reconstruction algorithms which are both effective and robust in the presence of errors. Exploiting the fact that each probe occurs at a unique point, we take it as our fundamental problem to determine the left-to-right order of the probes along the DNA strand. Once this order is known the overlap structure of the clones can easily be inferred.

All the central computational problems in this paper are NP-hard. For this reason we have evaluated the effectiveness of our algorithms empirically, using simulated data generated according to a probabilistic model, as well as real data from human chromosome 21.

1.2 The Maximum Posterior Probability Problem

In this section we formulate the problem of finding the most likely values of the underlying data and the ordering of the probes, given the measured data. The underlying data specifies the incidences between probes and clones; it also includes the information that certain probes have been extracted from the ends of particular clones; such a probe is called an *end probe* from the associated clone.

To define the problem we must specify the joint distribution of the following three random variables:

- $\mathcal{A}$, which ranges over all possible values A of the underlying probe-clone incidence data (see

	Clones							
	2	0	0	0	0	0	2	0
	1	1	0	0	0	0	1	0
P	2	1	2	2	0	0	1	0
r	0	2	1	1	1	2	0	0
o	0	0	1	1	1	1	0	0
b	0	0	0	2	2	1	1	0
e	0	0	0	0	0	1	1	2
s	0	0	0	0	0	2	1	1
	0	0	0	0	0	0	2	1
	0	0	0	0	0	0	0	1

The underlying data A
in correct order

	Clones							
	2	0	0	0	0	0	2	0
	1	1	0	0	0	0	1	0
P	2	0	2	2	0	0	1	0
r	0	2	1	1	1	2	0	1
o	0	0	1	1	1	1	0	0
b	0	0	0	2	2	1	1	0
e	0	0	0	0	0	1	1	2
s	0	0	0	0	0	2	1	1
	0	1	0	0	0	0	2	0
	0	0	0	0	0	0	0	1

The measured data D
in correct order

Figure 1: Example of underlying data matrix A and measured data matrix D

Figure 1). The matrix A has a row for each probe and a column for each clone; the i, j element a_{ij} is equal to 0 if probe i is not incident with clone j, to 2 if probe i is an end probe from clone j and to 1 if probe i is incident with clone j but is not an end probe from j. Each probe is an end probe from at most one clone and each clone has at most two end probes.

- $\mathcal{P}$, the correct probe ordering.
- $\mathcal{D}$, which ranges over all possible values D of the measured probe-clone incidence matrix. The matrix D has a row for each probe and a column for each clone. D differs from A because of measurement errors called *false positives* and *false negatives*. A false positive occurs when $a_{ij} = 0$ and $d_{ij} = 1$, where d_{ij} is the i, j element of D. Similarly, a false negative occurs when $a_{ij} = 1$ and $d_{ij} = 0$. Since end probes are extracted from known clones, we assume that this data is not subject to error; thus $d_{ij} = 2$ if and only if $a_{ij} = 2$.

The joint distribution of $\mathcal{A}$, $\mathcal{P}$ and $\mathcal{D}$ will depend on the following parameters:

- m, the number of probes;
- n, the number of clones;
- p, the chimeric clone rate;
- ϵ, the false negative rate;
- δ, the false positive rate;

The distribution also depends on the end probe data, which is arbitrary except for the constraints that each clone has at most two end probes and each probe is an end probe from at most one clone. Finally, it depends on the distribution of the number of probes on each clone.
The specification of the joint distribution is given in two parts:

1. The *a priori* distribution of the state of nature; i.e., the distribution of the ordered pair $(\mathcal{A}, \mathcal{P})$.
2. The conditional distribution of $\mathcal{D}$ given that $\mathcal{A} = A$ and $\mathcal{P} = \pi$. This distribution is determined by the false positive rate and false negative rate.

We sometimes use the symbols A, π and D to denote the events $\mathcal{A} = A$, $\mathcal{P} = \pi$ and $\mathcal{D} = D$, respectively. Using this convention, the goal of the maximum posterior probability method, given the measured data D, is to find a matrix A and a permutation π to maximize $p[A \wedge \pi | D]$. Equivalently,

the goal is to maximize the product $p[\pi] * p[A|\pi] * p[D|A \wedge \pi]$. We assume that all permutations of the probes have the same *a priori* probability, so that $p[\pi]$ is a constant. Thus the problem reduces to maximizing the product $p[A|\pi] * p[D|A \wedge \pi]$ over all $A \in \mathcal{A}$ and $\pi \in \mathcal{P}$.

We now specify $p[A|\pi]$. Let A^π denote the matrix obtained by permuting the rows of A into the order specified by π; thus $A^\pi_{i,j} = A_{\pi(i),j}$. Then it suffices to specify the distribution of A^π, assuming that π is the correct probe ordering. Each column of A^π corresponds to either a normal clone or a chimeric clone. In a column corresponding to a normal clone the ones form a consecutive block, since the probes incident with a normal clone must be consecutive in the ordering π. Also, each 2 in such a column must occur either first or last in the block, since it represents an end probe from the clone. Similarly, in a column of A^π corresponding to a chimeric clone, the ones form two blocks, each block may contain at most one 2, and a 2 must occur either first or last in its block. It remains to specify the distribution of block sizes. This distribution should reflect the distribution of clone lengths, clone locations and probe locations in real data. For example, if the clones are of constant length L and the probes occur according to a Poisson process of rate λ, then block sizes should have a Poisson distribution with mean λL. However, to avoid cumbersome calculations in our algorithm for maximizing $p[A \wedge \pi|D]$ we make the simplifying assumption that the distribution of block sizes is uniform over a range of integers $[minsize..maxsize]$, where $minsize$ and $maxsize$ are input parameters. Furthermore, we assume that the distribution of block sizes is insensitive to clone and probe location.

Given the identities of the end probes for each clone, our probability model makes the following assumptions about A^π:

- The clone corresponding to a column is chimeric with probability p, and normal with probability $1 - p$;
- If the column is normal then its endpoints (i, j) are drawn from the uniform distribution over all ordered pairs such that:
  1. $1 \leq i \leq j \leq m$;
  2. $minsize \leq j - i + 1 \leq maxsize$;
  3. the index of each end probe for the clone lies in the set $\{i, j\}$.
  4. the block of nonzeroes in the column extends from row i to row j, and twos occur precisely in those rows (if any) corresponding to end probes for the clone.
- If the clone is chimeric then the endpoints of its two fragments, (i, j, k, l), are drawn from the uniform distribution over all 4-tuples such that:
  1. $i \leq j < k \leq l$;
  2. $minsize \leq j - i + 1 \leq maxsize$;
  3. $minsize \leq l - k + 1 \leq maxsize$;
  4. the index of each end probe for the clone lies in the set $\{i, j, k, l\}$;
  5. if the clone has two end probes then the index of one of them lies in the set $\{i, j\}$ and the index of the other in the set $\{k, l\}$;
  6. the column has two blocks of nonzeroes, one extending from row i to row j, and the other from row k to row l, and twos occur precisely in those rows (if any) corresponding to end probes for the clone.

Finally, we specify $p[D|A \wedge \pi]$. This distribution depends on the *false negative rate* ϵ and the *false positive rate* δ. If a cell contains a 2 in A then it contains a 2 in D. If a cell contains a 0 in A

then it contains a 1 in D with probability δ and a 0 in D with probability $1-\delta$. If a cell contains a 1 in A then it contains a 0 in D with probability ϵ and a 1 in D with probability $1-\epsilon$.

Note that the matrix A^π is highly constrained: in each of its columns the nonzeroes occur in one or two consecutive blocks, and the twos occur at block boundaries. The number of ones in each block must lie between $minsize$ and $maxsize$. Let $\mathcal{A}^\pi$ be the set of matrices satisfying these constraints. Using Bayes' Theorem, one can show that the problem of maximizing $p[A \wedge \pi | D]$ is equivalent to the following: Choose a permutation π and a matrix A such that A^π lies in $\mathcal{A}^\pi$ and A has twos in exactly the positions where D has twos, to minimize

$$\left(\frac{2p}{(n-1)(1-p)s}\right)^{b_0(A,\pi)} \left(\frac{p}{(1-p)ns}\right)^{b_1(A,\pi)} \left(\frac{p}{(1-p)(2s-1)}\right)^{b_2(A,\pi)} \left(\frac{\epsilon}{1-\delta}\right)^{c(A,\pi)} \left(\frac{\delta}{1-\epsilon}\right)^{d(A,\pi)}$$

where:

- $s = maxsize - minsize + 1$;
- for $i = 0, 1, 2$, $b_i(A)$ is the number of columns of A which are designated as chimeric and have exactly i end probes;
- $c(A)$ is the number of cells containing a zero in D and a one in A;
- $d(A)$ is the number of cells containing a one in D and a zero in A.

The base in each of the first three factors is of the form $\frac{p}{(1-p)}\frac{N(i)}{C(i)}$, where $N(i)$ is the number of ways block boundaries can be assigned for a normal clone with i end probes, and $C(i)$ is the number of ways block boundaries can be assigned for a chimeric clone with i end probes.

Letting $K_0 = -\ln \frac{2p}{(n-1)(1-p)s}$, $K_1 = -\ln \frac{p}{(1-p)ns}$, $K_2 = -\ln \frac{p}{(1-p)(2s-1)}$, $L = -\ln \frac{\epsilon}{1-\delta}$ and $M = -\ln \frac{\delta}{1-\epsilon}$, we can re-express the maximum posterior probability problem as follows: choose π and A^π in $\mathcal{A}^\pi$ to minimize $K_0 b_0(A,\pi) + K_1 b_1(A,\pi) + K_2 b_2(A,\pi) + Lc(A) + Md(A)$. Section 3.1 describes an effective algorithm to find a good solution of this optimization problem. The problem is NP-complete even in the absence of false positives and chimeric clones [B75].

2 Algorithmic Approaches

We present several approaches to the problem of recovering the underlying data A and the permutation π from the measured data D.

2.1 The Maximum Posterior Probability Method

As discussed above, we attempt to find a matrix A and a permutation π which maximize $p[A \wedge \pi | D]$. Given a particular π, we can quickly determine an optimal A. We use simulated annealing to search over the space of π's. In Section 3.1, we present an effective procedure for doing this.

2.2 The Hamming Distance TSP

We describe this approach in the case where end clone information is not available, so that the matrix A of underlying data contains only zeroes and ones.

There exists a permutation π of the rows of A which produces a matrix A^π in which each column corresponding to a normal clone contains one block of ones, and each column corresponding to a chimeric clone contains two blocks of ones. Define a *gap* as a block of zeroes in a column, flanked

by a one immediately above and a one immediately below. Then the number of gaps in A^π is equal to the number of chimeric clones. Now consider the effect of introducing random errors. A false negative will change a one to a zero, typically splitting a block into two parts, and thus creating a gap. A false positive will change a zero to a one, typically splitting a gap into two gaps. Thus, when the permutation π is applied to the matrix D of measured data, the number of gaps in the resulting matrix D^π tends to be approximately equal to the number of chimeric clones plus the number of false positives and false negatives. This suggests the heuristic principle that a permutation of the rows which minimizes the number of gaps will correspond to a good probe ordering. Minimizing the number of gaps can be cast as a Traveling Salesman Problem called the Hamming Distance TSP [AKNW91], in which the cities are the rows of A together with an additional row of n zeroes, and the distance between two cities is the Hamming distance between the corresponding rows; i.e., the number of positions in which the two rows differ. The Hamming distance TSP is NP-complete by transformation from the Hamiltonian path problem.

The Hamming Distance TSP has the disadvantage that its objective function is insensitive to the false positive rate, the false negative rate and the frequency of chimeric clones; thus it models the choice of an optimal probe ordering less faithfully than the maximum posterior probability approach. Its advantage is that the technology for solving large Traveling Salesman problems is far advanced, so that near-optimal solutions are very easily obtained. We use a local search method due to Zweig [Z92], based on an operation called Divide-and-Merge, which rapidly and dependably gives near-optimal solutions to instances of the Hamming Distance TSP with around one thousand cities. We have found that, with a coverage of 10 or more and at least as many probes as clones, near-optimal solutions to the Hamming Distance TSP indeed correspond to near-optimal probe orderings. This is true over a range of error rates (see Computational Results).

2.3 Methods for Obtaining a Good Initial Probe Ordering

The maximum posterior probability approach leads to a complex optimization problem which we attack by a local search method (cf. Section 3.1). The execution time of the local search can be reduced by providing the algorithm with a good initial solution. In addition to using the TSP solution, we have explored two other methods of obtaining such an initial solution: sorting and splitting.

2.3.1 Sorting

In the absence of false positives, false negatives and chimeric clones, the true ordering of the probes satisfies the following *ordering principle*. Let C_i be the set of clones incident with probe i. If probe j lies between probes i and k in the true ordering, then $|C_i \cap C_j| \geq |C_i \cap C_k|$ and $|C_j \cap C_k| \geq |C_i \cap C_k|$. We have devised a fast method called *sorting* which iteratively rearranges the probes with the goal of finding a probe ordering satisfying the ordering principle. Such a procedure must necessarily be imperfect since, in the presence of errors and chimeric clones, there may not exist an ordering that satisfies the ordering principle. The method is presented in Section 3.2. It has been found to give a good initial ordering, thereby reducing the time required for the maximum posterior probability calculation.

2.3.2 Splitting

Let G be a bipartite graph with vertex set $\mathcal{P} \cup \mathcal{C}$, where $\mathcal{P}$ is the set of probes and $\mathcal{C}$ is the set of clones, and with an edge between probe p and clone c if and only if p is incident with c. We

assume that G is connected, as otherwise the physical mapping problem decomposes into connected components whose order along the chromosome cannot possibly be ascertained from the data.

We say a set of probes X *covers* a probe p if each clone incident on p is also incident on some probe in X. Briefly, given a probe p, we remove the set of probes P that share a clone with it, all the probes covered by P, and all the clones that are not incident with any of the remaining probes. If G is left with exactly two components then p is a *splitter*. In this case we accept that the probes in one of these subgraphs lie to the left of p, and the probes in the other lie to the right of p. When there are no false positives or chimeric clones the ordering implied by all the splitters is good, and its use as an initial ordering speeds up the maximum posterior probability computation.

2.4 Screening and Post Screening

We have found that false positives and chimerics are particularly troublesome for our algorithms. Therefore, we have devised methods for removing false positives and splitting chimeric fragments into their component parts. These methods are not foolproof, and they sometimes destroy linkages between nearby probes; however, the overall effect of the screening methods is quite favorable.

By removing probe-clone incidences that are incorrectly labeled as false positives and splitting clones that are not chimeric, the screening process can sometimes incorrectly break a contig into several components. Without restoring these critical linkages, it is not possible to determine the order in which these components occur along the DNA strand. Thus, we use a two stage approach. In the first stage we obtain a good ordering for the probes in each connected component. Then, by inspecting the original data, we restore critical information that hooks the components together. This process is called *post screening*.

Screening is described in Section 4.1 and post screening is described in Section 4.2.

2.5 Algorithmic Strategy

We have experimented with a number of strategies for finding a good probe ordering. A very fast method which gives reasonably good solutions is simply to solve the Hamming Distance TSP. The following strategy has proved effective for obtaining better solutions, at the cost of a modest increase in execution time.

- Apply screening to eliminate false positives and split chimeric clones.
- After screening the data may decompose into several connected components. For each connected component:
    1. Use the sorting heuristic or the Hamming distance TSP to obtain a reasonably good initial probe ordering.
    2. Starting with this initial ordering, obtain a near-optimal solution to the maximum posterior probability problem.
- Use a post screening procedure to hook together the probe orderings for the several connected components of the screened data.
- This strategy can be enhanced by a retesting process in which the results of the optimization procedure are used to identify likely experimental errors. The corresponding probe-clone incidences can then be rechecked in an iterative fashion.

Our computational results are summarized in Section 7.

2.6 Relationship to Previous Work

In the field of physical mapping, simulated annealing has been used by [RML93], [AGL94], and [ACT93], among others. [RML93] presents a software package used to construct a YAC map of *S. pombe*. This package uses simulated annealing to minimize a distance measure, $d_{ab} = \frac{n_{ab} - r_{ab}}{n_{ab}}$, summed over consecutive pairs of probes a and b. n_{ab} is the number of clones hybridizing to either probe, and r_{ab} is the number of clones hybridizing to both. That paper also presents a screening strategy that is similar to ours except that it discards chimeric clones rather than splitting them; [AGL94] presents an updated screening procedure that splits chimeric clones.

[ACT93] presents a simulated annealing approach to ordering clones by minimizing the Hamming distance summed over consecutive pairs of clones. The Hamming distance objective function is also used to order clones in [YWA94]. A heuristic search procedure based on random clone reversals is described, and found to outperform a simulated annealing implementation. The Hamming distance metric has seen extensive use in radiation hybrid mapping; [MBC91] provides a comparison with other methods in that context.

[GI94] presents two algorithms for ordering probes when the data is contaminated with chimeric fragments, but not false positives or false negatives. The first of these approaches is based on minimizing χ, the maximum number of chimeric fragments imputed on a single clone, while the second approach is based on minimizing σ, the total number of chimeric fragments summed over all clones.

The maximum posterior probability objective function we use is an improvement on the methods described previously because it is sensitive to the relative rates of false positive, false negative, and chimeric errors. Furthermore, it can be extended in a principled way to accommodate other errors such as deletions. Our retesting results indicate that this objective function is highly effective in identifying errors.

3 Reconstruction Algorithms

3.1 Local Search

Recall that the choice of A involves specifying a linear ordering π of the rows of D, designating each column as normal or chimeric, choosing a block of rows for each normal column, and choosing two blocks of rows in each chimeric column. We shall show that, given π, there is a simple procedure for making the remaining choices optimally. The resulting optimal choice of the underlying data given π will be denoted $A(\pi)$. In the determination of block boundaries, we do not enforce the restriction that block sizes lie between *minsize* and *maxsize*. While these restrictions provide appropriate values for the *a priori* probabilities, they approximate what in reality is an unbounded distribution. Therefore we do not bias against blocks with unusual lengths; this has the added advantage that no blocks that begin or end with zeros need be considered. This yields the following optimization problem: Find π to minimize the objective function $F(\pi)$, defined as $K_0 b_0(A(\pi), \pi) + K_1 b_1(A(\pi), \pi) + K_2 b_2(A(\pi), \pi) + Lc(A(\pi), \pi) + Md(A(\pi), \pi)$. K_0, K_1, K_2, L, and M will be referred to as the "costs" of chimerics (with the specified number of end probes), false negatives, and false positives respectively. Our strategy for solving this optimization problem is based on the fact that for a given π, $A(\pi)$ and $F(\pi)$ are easily computed.

$F(\pi)$ is the sum of terms derived from the individual columns of the matrix. To determine the contribution of each column, we make two simple calculations, one based on the assumption that the column is normal, and the other on the assumption that the column is chimeric. The contribution of a column to $F(\pi)$ is then the smaller of the two terms. Let C be the number of nonzero entries in the column; in typical cases C is a small constant. If the column is normal, then we need to choose

the first and last rows in its block. For an optimal choice, these rows must contain ones or twos in the given column (given the relaxed assumptions on block lengths). Thus, there are $\binom{C}{2}$ choices for the block boundaries, but by using appropriate data structures and dynamic programming, the optimal boundaries can be determined in time $O(C)$, as described in Section 3.1.1.[1] If the column is chimeric we must choose the boundaries of two blocks, and we may assume that the first and last row of each block contain ones or twos. Thus there are $\binom{C}{4}$ choices for the block boundaries, but again the optimal choices can be determined in linear time. These times hold both with and without end probe information.

3.1.1 Computing $A(\pi)$

The following paragraphs describe our $O(C)$ method for the objective function calculation in more detail. The procedure commences by replacing consecutive occurrences of zeros or ones with single symbols. Attached to each symbol is the cost of negating all the underlying zeros or ones, i.e. of implying that they are all false positives or negatives. This simplification relies on the fact that it cannot be advantageous to assign a block boundary in the middle of a sequence of zeros or ones.

Two scans are then made over the symbols, one from left to right, and the other from right to left. Only the left to right scan will be described here as the other is analogous. The purpose of this scan is to create an array, *mlr*, whose *ith* element lists the maximum "advantage" that can be gained by positioning the main block of ones to the left of the *ith* symbol (possibly including it). "Advantage" refers to the advantage in cost that is gained over the cost of implying all the positive probe-clone incidences to be experimental errors, which we call the total negation cost. Thus, in the absence of twos, and assuming the clone is not chimeric, the optimal cost is given by the total negation cost, less the value stored in the last entry of the array. Assuming that the column is chimeric, we need both the left-to-right and right-to-left arrays, and search for the position i for which $mlr[i] + mrl[i+1]$ is maximized, and the optimal cost is the total negation cost less this amount, plus the appropriate cost for implying chimerism.

The presence of twos is accommodated by initializing the arrays to zero advantage each time a two is encountered. After these arrays are initialized, we calculate the optimal cost for each possible endpoint configuration of the twos, and choose the best. Consider, for example, a clone that we assume to be chimeric, in which two twos are present. Assuming the twos are in positions $t1$ and $t2$, and that they mark the leftmost endpoints of the two fragments, and that the total number of symbols is k, the maximum advantage is given by $mlr[t2-1] + mlr[k]$. Assuming the first two is a rightmost endpoint and the second two is a leftmost endpoint, the optimal cost is given by $mrl[1] + mlr[k]$. Similar calculations can be made for all the other possibilities.

Pseudocode is presented below that outlines the generation of the left to right array, *mlr*. An auxiliary array, *partial*, is also used. Note that the advantage of having a two (a known endpoint) in a block is set to zero rather than infinity; the constraint is enforced after the arrays are initialized.

Array Initialization:

```
partial[0] = 0;
mlr[0] = -HUGE_AMOUNT;
boolean seen_a_two = false;
for (unsigned i=1; i<=Num_Symbols; i++)
```

[1] In a previous paper we described an $O(C^2)$ algorithm for column evaluation. We are grateful to Mudita Jain and Eugene Myers for originally developing a dynamic programming approach to this problem, thus reducing the running time to $O(C)$.

```
{
    if (Symbol[i].type == 0)
        ad = -Symbol[i].negation_cost;
        // ad is the advantage of having the symbol in a block
    else if (Symbol[i].type == 1)
    {
        ad = Symbol[i].negation_cost;
        Total_Negation_Cost += ad;
    }
    else // a 2
    {
        seen_a_two = true;
        ad = 0;
    }

    // calculate the partial sum and max if appropriate.
    partial[i] = partial[i-1] + ad;
    if (!seen_a_two)
    {
        if (ad > partial[i])
            partial[i] = ad;
    }
    else if (Symbol[i].type == 2)
        partial[i] = mlr[i] = 0; // the entry is a 2; start over.
    if (Symbol[i].type != 2)
        if (mlr[i-1] > partial[i])
            mlr[i] = mlr[i-1];
        else
            mlr[i] = partial[i];
}
```

3.1.2 Neighborhood Structure

We attack the problem of optimizing over permutations by local search, using a move operation which is related to the 2-opt and Or-opt move operations used for the Traveling Salesman Problem. Given the present permutation π, the move operation selects a "neighboring permutation" as follows. Let D^π be the matrix obtained by reordering the rows of D in accordance with the current permutation; i.e., $D^\pi_{i,j} = D_{\pi(i),j}$. Define a *gap* in column j as an interval $[i..k]$ in column j, such that rows i and k contain ones in column j, and the intervening rows contain zeroes.

Operation Move:

- Select a gap at random in the matrix D^π. Let it be interval $[i..k]$ in column j. The chosen move will make rows i and k adjacent.
- Let u, the *upper weight* of the gap, be defined as the number of ones in column j at or above row i, and let ℓ, the *lower weight* of the gap, be the number of ones in column j at or below

row k. Randomly choose either to move row i or to move row k, where the probability of choosing i is $\frac{\ell}{\ell+u}$ and the probability of choosing k is $\frac{u}{\ell+u}$;

- If i has been chosen then either move row i downward past rows $i+1, \cdots, k-1$ or reverse the block of rows $i, i+1, \cdots, k-1$, the two choices being equally likely;
- If k has been chosen then either move row k upward past rows $k-1, \cdots, i+1$ or reverse the block of rows $i+1, i+2, \cdots, k$, the two choices being equally likely.

We have observed that, in cases where the only errors are false negatives, the move operator appears to have the following property: if the move operator is applied repeatedly, then after a long enough period of time the successive permutations oscillate around a near-optimal permutation; i.e., there is a near-optimal permutation τ such that, for each probe i, the long-run average position of probe i approaches $\tau(i)$. In the case of noiseless data, random iteration of the move operation tends to converge to an ordering with the consecutive ones property. We have no theoretical explanation for these phenomena, (see Figure 17.)

Since the above move is defined only in terms of gaps, and makes no reference to known end probes, it sometimes happens that its application fails to place twos in extremal positions. Therefore we occasionally apply another move, in which a non-extremal two is chosen at random, and the row in which it lies is swapped with another row chosen at random in which there is a one in the same column as the selected two. Typically this move is used ten percent of the time.

3.1.3 Simulated Annealing Strategy

We have found that simulated annealing, using the move operators defined above, is an effective method of obtaining near optimal solutions. The simulated annealing strategy that we use, however, is unusual in that we typically start from a good solution. In order to take advantage of this fact, we have developed an adaptive way of setting the initial temperature. It is based on the observation that if a solution is near optimal, random moves will almost never increase its quality, whereas if the solution is random, random moves will improve it about half the time.

If we start with a random solution, we desire that essentially all the moves that are initially proposed be accepted. On the other hand, if we start with an optimal solution, we desire that no moves be accepted. The heuristic we adopt produces this behavior by setting the initial temperature to a value that accepts a fraction of the moves equal to twice the fraction of random moves that cause improvement. Other than in setting the initial temperature, we use a conventional cooling schedule and a conventional stopping rule [L88] [AL87]. The maximum number of moves considered at each temperature step is 20 times the number of STSs. The temperature is multiplied by 0.95 as soon as this number of moves have been attempted, or when 25% of this number have been accepted. The procedure terminates when less than 5% of the moves are accepted for four consecutive temperature steps.

In its relation to simulated annealing, the neighborhood structure that we use is significant for several reasons. First it is important to note that the moves are not in general reversible; for example, when a gap is eliminated, there is no guarantee that it can be re-created. This is important because the theoretical justifications of simulated annealing are predicated on a symmetrical transition matrix. The non-reversible moves that we use have the advantage, however, that in the absence of false positives and chimerics, convergence is much faster than it would otherwise be; excellent solutions are often found early on despite the fact that almost all the moves that are proposed are accepted. It is possible to take advantage of this by terminating the search when the best solution found is not replaced within a limited number of moves. Computational experience with the simulated annealing algorithm is reported in Section 7.

3.2 Sorting

We have devised a simple procedure for finding a good initial ordering of the probes. The procedure is based on refinements of the following *ordering principle*. Let the set of probes be $\{1, 2, \cdots, m\}$. Let C_i be the set of clones incident with probe i.Then, in the absence of errors, we may make the following inference: if probe j lies between probes i and k in the true ordering of the probes, then $|C_i \cap C_j| \geq |C_i \cap C_k|$ and $|C_j \cap C_k| \geq |C_i \cap C_k|$. The ordering principle is rigorously valid in the absence of noise and chimeric clones, since every clone that is incident with both probe i and probe k must also be incident with probe j. In the presence of noise and chimeric clones the principle may fail, but it remains a good heuristic guide to the construction of a good probe ordering. We have devised a *sorting heuristic* that seeks to construct a good probe ordering by iteratively permuting the probes with the goal of enforcing the ordering principle.

We make the following assumptions:

Connectedness For every partition of the set of probes into two nonempty sets A and B, there exist probes $i \in A$ and $j \in B$ such that $C_i \cap C_j \neq \phi$;

Distinguishability If $i \neq j$ then $C_i \neq C_j$.

There is no essential loss of generality in these assumptions; if the connectedness assumption is violated then the set of probes divides into noninteracting connected components whose order cannot be ascertained from the data; if the distinguishability assumption is violated then each set of indistinguishable probes can be treated as a single probe for the purpose of probe ordering.

3.2.1 The Noiseless Case

We begin by considering the *noiseless case*, in which there are no false positives, false negatives or chimeric clones. The goal in this case is to find a probe ordering such that probes incident with each clone are consecutive. The PQ-tree algorithm of Booth and Lueker computes a succinct representation of all such orderings in linear time. Here we give some alternative approaches based on the ordering principle. The next lemma, whose simple proof we omit, shows the equivalence of the consecutive ones property and the ordering principle.

Lemma 1 *Let $\sigma = (\sigma(1), \sigma(2), \cdots, \sigma(m))$ be a permutation of the probes. Then, in the noiseless case, the following are equivalent:*

Consecutive Ones Property *The probes incident with each clone are consecutive in σ;*

Ordering Principle *If $i < j < k$ then $|C_{\sigma(i)} \cap C_{\sigma(j)}| \geq |C_{\sigma(i)} \cap C_{\sigma(k)}|$ and $|C_{\sigma(j)} \cap C_{\sigma(k)}| \geq |C_{\sigma(i)} \cap C_{\sigma(k)}|$*

Define the *noninclusion property* as follows: for every ordered pair (C_1, C_2) of clones, there is a probe that is incident with C_1 but is not incident with C_2. When this property holds the probe-clone incidence data determines the true probe ordering uniquely up to reversal, and we can characterize the true ordering using a refinement of the ordering principle.

For each probe i, define the following partial ordering $\succ_i$ of the set of probes: $j \succ_i k$ if

$$(|C_i \cap C_j| > |C_i \cap C_k|) \vee (|C_i \cap C_j| = |C_i \cap C_k|) \wedge (C_i \cap C_j \neq \phi) \wedge (|C_j| < |C_k|).$$

This partial order has the following significance when the data is noiseless and the noninclusion property holds: If $j \succ_i k$ then probe k does not lie between probe i and probe j in the true ordering. Let $N(i)$ be the set of probes that occur together with probe i on at least one clone. Formally, $N(i)$ is the set of probes j such that $C_i \cap C_j \neq \phi$; in particular, probe i lies in $N(i)$.

Lemma 2 *Suppose that*

- *the connectedness and distinguishability assumptions hold;*
- *there are no false positives, false negatives or chimeric clones, and*
- *the noninclusion property holds.*

Then the true ordering of the probes is uniquely determined up to reversal by the requirement that the following properties hold for each i:

- *The probes in $N(i)$ occur consecutively in the ordering as a block $B(i)$;*
- *Starting at probe i and moving either to the right or to the left, the probes in $B(i)$ form a decreasing chain in the partial order $\succ_i$.*

We present a simple algorithm that finds the true ordering when the hypotheses of Lemma 2 hold. Throughout the algorithm the variable π denotes a sequence of distinct probes from the set $\{1, 2, \cdots, m\}$, α denotes the first element of π and ω denotes the last element of π. Throughout the process, the sequence π constitutes a consecutive block in the true ordering. At the beginning of the first step, π is initialized to the sequence consisting of the single probe 1. Each step adjoins one element to π as follows:

1. If every element of $N(\omega)$ lies in π then replace the sequence π by its reversal.
2. Choose a probe k which, among probes in $N(\omega)$ which do not occur in π, is greatest in the ordering $\succ_\omega$. If $\omega \succ_k \alpha$ then append k to the end of π; otherwise, append k to the beginning of π.

Using Lemma 2 one can prove inductively that, after every step, the probes in π form a consecutive block in the true ordering. Since a new element is adjoined to π at each step, it follows that, after the mth step, π gives the true ordering of all the probes.

3.2.2 The Sorting Algorithm

We next present an algorithm for ordering the probes in the presence of errors and chimeric clones. It is motivated by the algorithm for finding the true ordering of the probes in the noiseless case under the noninclusion assumption, and in that case it produces the true ordering. It is based on the same s $\succ_i$ and sets $N(i)$ used in that algorithm. However the sorting algorithm is more elaborate than that algorithm, since we would like it to produce a reasonably good ordering even in the presence of errors, chimeric clones and proper inclusion of one clone by another.

The sorting algorithm is guided by the desire to order the probes so that, for each i,

- The probes in $N(i)$ occur consecutively in the ordering as a block $B(i)$;
- Starting at probe i and moving either to the right or to the left, the probes in $B(i)$ form a decreasing chain in the partial order $\succ_i$.

In general, such an ordering will not exist, so the heuristic will be forced to make some compromises.

Our algorithm for ordering the probes proceeds in m steps. Throughout the algorithm the variable π denotes a sequence of distinct probes from the set $\{1, 2, \cdots, m\}$, α denotes the first element of π and ω denotes the last element of π.

At the beginning of the first step, π is initialized to the sequence consisting of the single probe 1. Each step adjoins a new element to π and rearranges π in accordance with the ordering principle. At the end of the process π will contain all the probes.

Each step proceeds as follows:

- If every element of $N(\omega)$ lies in π then replace π by its reversal;
- Choose a probe k which is maximal in the ordering $\succ_\omega$ among the probes which do not lie in π;
- Append k to the end of π;
- Sort π into the unique linear order such that:
  1. If $r \succ_k s$ then s precedes r;
  2. If r and s are incomparable in $\succ_k$ then they maintain the same relative order as before the step.

We have found that the performance of the sorting algorithm can be improved in practice by introducing a parameter w called the *window size* which limits the effect of any given iteration; instead of sorting the entire sequence π at each step, only the last w elements of π are sorted. The performance of the algorithm is also improved by dividing the computation into two phases, where the purpose of the first phase is to find a probe which will occur first in the final ordering. This is done by running the above algorithm until the first time that every element of $N(\omega)$ lies in π. The probe ω (i.e., the last probe in π) is then identified as the first probe in the final ordering, and the computation is restarted with π initialized to the sequence consisting of the single probe ω. Section 7 reports on the computational results obtained using a version of the sorting algorithm which incorporates these two improvements.

3.3 Splitting

Splitting is a method for finding an approximate probe ordering. It works well in the presence of false negatives, but becomes ineffective when false positives or chimeric clones are present. Splitting works by repeatedly breaking the contig into right and left components, and counting the number of times each probe appears on either side. Consider the bipartite graph G as described in Section 2.3.2. There is a vertex for each clone and probe, and an edge for each clone-probe incidence. We assume that this graph is connected, as otherwise the mapping problem decomposes into disconnected components whose order along the chromosome cannot be ascertained from the data.

We say a set of probes P *covers* a probe p if each clone incident on p is also incident on some probe in P. Let $N(s)$ be the set of probes that share a clone with probe s. Probe s is called a *splitter* if, when $N(s)$ and all the probes covered by $N(s)$ as well as those clones that are not incident with any of the remaining probes are removed, the resulting subgraph of G has exactly two connected components. Let the vertex sets of these two components be denoted by A and B, and let the vertices in neither component be denoted by M. In the noiseless case, M correctly separates A from B; i.e., one of the two possible left-to-right orientations of the DNA strand has the property that every probe or clone in A is strictly to the left of every probe or clone in M, and every probe or clone in B is strictly to the right of every probe or clone in M.

In the case where false negatives occur but false positives and chimeric clones do not, the partitions associated with splitters still provide valuable information about the ordering of the probes. We have found empirically that a probe which is a splitter with respect to the noiseless data and is not involved in any false negatives is also likely to be a splitter with respect to the noisy data.

The above observations suggest that the following algorithm should give a good probe ordering in the absence of false positives and chimeric clones.

1. Determine which probes are splitters and, for each splitter, determine the components A and B;

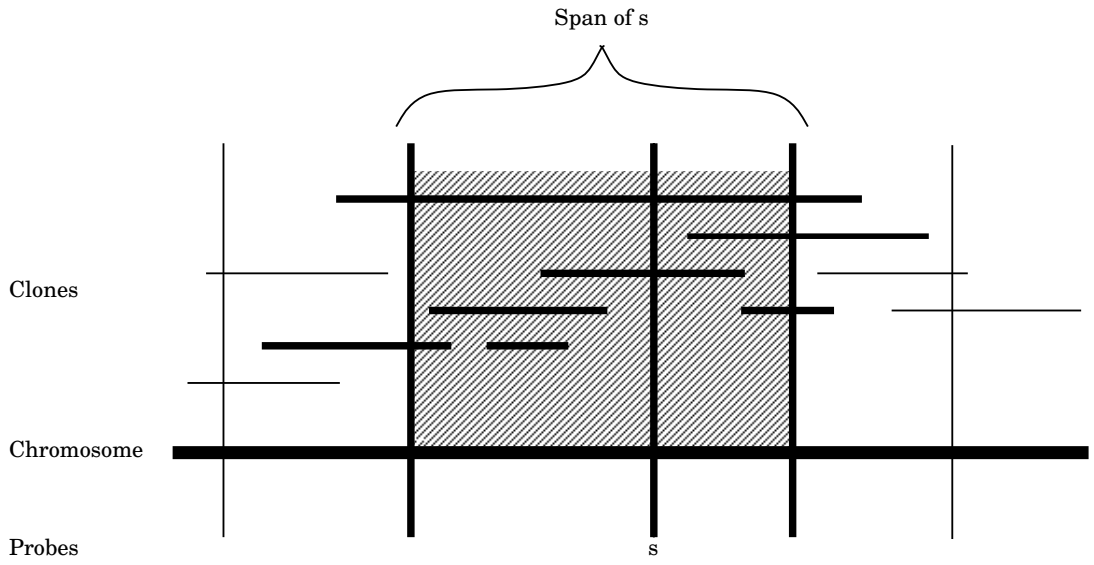

Figure 2: Probe s is a splitter. Darkened probes and clones are removed.

2. Arbitrarily select two sets of probes which occur in opposite components. Label one set "Left" and the other "Right".

3. Associate with each probe q a *left tally* $\alpha(q)$ giving the number of times it occurs in a component where a larger fraction of the probes intersect the "Left" than the "Right". Generate a *right tally* $\beta(q)$ analogously.

4. Sort the probes in decreasing order of $\alpha(q) - \beta(q)$, breaking ties in increasing order of $\alpha(q)$.

This procedure gives a good initial ordering, even when there are moderate number of false negatives.

4 Screening and Post Screening

4.1 Screening False Positives and Chimerics

As mentioned earlier, false positives and chimerics are especially troublesome for our algorithms. In particular, these types of errors degrade the speed of simulated annealing and render the splitting algorithm inoperable through lack of splitters. Moreover, the accuracy of all the algorithms is harmed. Therefore we have developed several methods for finding and removing false positive and chimeric probe-clone incidences, the best of which we call the spanning graph screener.

Let c be a clone and S the set of probes that are incident on it in the measured data. Two probes in S are said to belong together if neither is a false positive, and in case c is chimeric the two occur on the same fragment of c. The spanning graph screener attempts to partition S into parts each consisting of a maximal group of probes that belong together on c. Thus each false positive should constitute a part by itself, and if c is chimeric the remaining parts should correspond to the fragments of c. The spanning graph screener is based on the observation that if two probes in S

occur together on sufficiently many other clones, then they are likely to belong together on c. It works as follows:

- Set a threshold level T.
- For each clone:
  - For each pair of probes on the clone create an edge whose weight is the number of other clones on which the two probes appear together.
  - Delete all edges with weight less than T.
  - Find the connected components of the graph formed by the edges.
  - Replace the clone with a separate "fictitious" clone for each component. This operation is called a *split*.

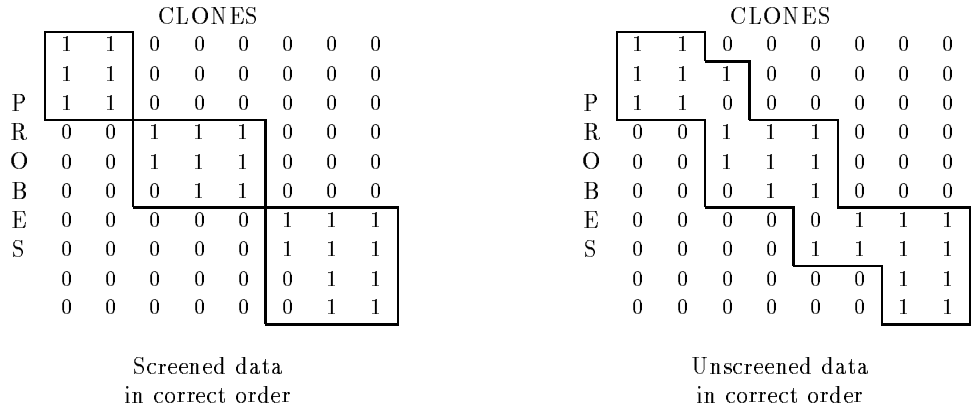

Figure 3: Disjoint connected components created by screening and reconnected by post screening

Figure 4 shows the performance of the screening algorithm for several combinations of chimeric and false positive rates.

False Positive Rate	cr = 0	cr = 0.25	cr = 0.5
0.1%	- , 94 , 73	95, 95, 38	94, 91, 14
0.5%	- , 92, 36	97, 88, 16	92, 87, 7
1.0%	-, 90, 21	92, 86, 8	88, 82, 6

200 clones, coverage 10, probe rate 1.0, and threshold 6. Chimeric rate (cr) varies between 0 and 0.5. False negative rate fixed at 10%. Each cell shows the percentages of chimerics and false positives that were screened, and the percentage of splits that did not correctly identify a false positive or a chimeric fragment. Percentages averaged over 10 runs.

Figure 4: Screening Results

4.2 Post Screening

The screening procedure will sometimes destroy correct linkages between probes, and thus break the contig into several smaller pieces. To counteract this, we have developed an algorithm which

identifies connected components created by the screening process and reorders them according to the unscreened data (see Figure 3.)

The post screening algorithm first finds connected components in the ordered, screened data. Since the rows are ordered, we assume that adjacent or nearly-adjacent rows which do not overlap lie in separate components.

Once determined, the components are reordered so as to maximize the average pairwise intersection between the probes near the ends of adjacent components. We use a greedy merging strategy that repeatedly links together the two components whose ends have the highest average intersection. When there are only a few components, an exhaustive search is possible, but we have found that greedy merging is a robust competitor that can function even when there are numerous components.

Results of the post screening algorithm are shown in Section 7.

5 Combining Permutations

It often happens that many very different permutations achieve optimal, or nearly optimal, objective function values. In such cases, the most useful information is found in the subsequences that all such permutations have in common, and we have developed a fast, robust way of extracting this information.

The user sets a threshold level t, and the pairs of probes that are within t positions of each other in every permutation are deemed to be adjacent. An adjacency matrix is constructed, and the connected components are found. Then the ordering of an arbitrarily chosen one of the given permutations is imposed on each component.

This scheme has several virtues. The first is its simple parameterization on the threshold level. A threshold level of one ensures that the extracted subsequences are identical in all permutations. Setting the threshold level higher allows for local discrepancies in the subsequences, and is correspondingly more robust. The scheme can also deal with an arbitrary number of permutations, and the calculations can be done incrementally and on-line as they are generated.

6 Retesting

We have developed a highly effective approach to recovering the noiseless data through an interactive process in which an algorithm is used to detect anomalies in the data, and these anomalies are then rechecked through further probe-clone incidence experiments. At a general step, for each probe i and clone j, some number $t_{ij} \geq 1$ of independent tests will have been conducted, each purporting to measure whether probe i occurs on clone j; let y_{ij} denote the number of such tests with a positive outcome, and let n_{ij} denote the number of such tests with a negative outcome. The problem of finding a maximum posterior probability matrix A of noiseless data, given the matrices (y_{ij}) and (n_{ij}) can be cast as a generalization of the minimization problem in Section 3.1. Near-optimal solutions to this problem can be found by local search. Once a solution $A = (a_{ij})$ has been found, we identify certain *questionable cells* whose data needs to be confirmed or disconfirmed by further tests. A matrix entry a_{ij} is deemed to be questionable if the maximum posterior probability model disagrees with the most recent test performed on the cell. Thus a cell remains questionable only until retesting confirms the algorithm's prediction. This allows highly improbable errors that are correctly identified by the optimization procedure to be corrected immediately, without the multiple retesting events that would otherwise be required. The identification of the underlying data then proceeds by repetition of the following algorithm:

- For the given data (y_{ij}) and (n_{ij}), use local search to find a matrix A and a permutation π that (nearly) maximizes $p[A \wedge \pi | D]$.

- Identify and retest the questionable cells.

The iteration continues until no questionable cells remain. Because the false positive and false negative costs are different for different probe-clone incidences, an added sweep between the extremal nonzero entries of each column is necessary in the column evaluation procedure.

Using simulated data, only a few iterations are typically required to identify the correct permutation of the probes, and the number of repeated tests is comparable to the number of errors present in the data.

Figure 5 presents the number of retests required to obtain internally consistent data using the method described in Section 6.

False Positive Rate	Ratio of Retests to Original Errors	Ratio of Retests to Correctly Identified Errors
0.1%	1.16	1.29
0.5%	1.12	1.19
1.0%	1.22	1.27

Average of five instances: 200 clones, coverage 10 and probe rate 1.0. False negative rate fixed at 10%

Figure 5: Retesting Table

7 Computational Results

In this section we present some computational results for the various algorithms. We present results using data from human chromosome 21 generated by Cohen's group at CEPH [CRG+92] as well as simulated data.

In the simulated data, unit length clones are generated so that their midpoints are distributed uniformly and independently over an interval of length N. [2] With probability p, a random, non-overlapping unit length fragment is generated and appended to a clone, thus creating a chimeric clone. Probes are chosen from the endpoints of clones. The probe rate $\rho \in [0, 2]$ is the expected number of probes taken from a single clone. In the case where $\rho \leq 1$, each clone is chosen with probability ρ to contribute a single probe. For $\rho > 1$, each clone contributes at least one probe, and with probability $\rho - 1$ contributes two probes. False negatives and false positives are introduced with rates ϵ and δ, as described in Section 1. The ratio of the number of clones to the length of the chromosome is called the *coverage*. Our results can be summarized by the following observations:

- The algorithms perform well when the coverage is at least seven, and extremely well when the coverage is at least ten.
- The splitter algorithm runs quickly and produces good permutations even when the false negative rate is high (see Table 9 and Figure 16), but it performs poorly in the presence of even a modest number of false positives and chimeric clones.
- The Hamming distance TSP runs quickly, but the permutation quality suffers when the false negative rate is high.
- Screening speeds up the running time of simulated annealing by a factor of approximately 20, but when the error rate is high, the final permutation may not be as good.

[2] Our algorithms also work when the clone sizes vary, in particular when one clone may be contained in another.

- Simulated annealing is by far the slowest algorithm, but it is also the most robust when the error rates are high.

In all of our simulations, the correct probe ordering is $[1, 2, \cdots, m]$. We present the computed permutations plotted against this identity permutation. In other words, we present the data in the form of a two-dimensional array, where a dot in the i, j entry denotes a probe in the ith position of the identity permutation and the jth position of the computed permutation. In the plots for the Chromosome 21 data, the results of our algorithms are plotted against the order published in [CRG+92]. The quality of an ordering can thus be judged by how close it is to the identity permutation or its reverse. Plots with long line segments of ± 45 degrees within connected components are more desirable than those with short scattered segments within components.

It sometimes happens that the STSs fall into several disconnected groups, and the interrelationships between those groups cannot possibly be discovered. The boundaries of such groups are marked on our graphs by horizontal lines. This situation occurs if, after false positives and chimeric fragments are correctly separated, the data breaks into several components. Obviously this check can only be done for simulated data.

In the first four sets of data, we show six plots and six sets of timings, one for each of the following algorithms and sequences of algorithms:

- Sorting
- Splitting
- Screening→Hamming Distance TSP→Simulated Annealing→Post Screening
- Screening→Sorting→Simulated Annealing→Post Screening
- Hamming Distance TSP, and
- Simulated Annealing.

The first set of results is on the Chromosome 21 data. In the second set we use simulated data and vary the coverage from 10 to 5. The third set of results is on simulated data where the false negative rate is 20% or 30%. The fourth set is on simulated data with a 30% false negative rate and no false positives or chimeric clones.

The final set of data helps substantiate our choice of move operator for the Local Search algorithm. We present plots of the best ordering found in $40,000$ random moves, and the average ordering from moves $35,000$ to $40,000$. We present results for the Chromosome 21 data and for simulated data with false negatives only. In the plots of average positions, a dot in the i, j entry denotes a probe in the ith position of the identity permutation, and whose average position over the last $5,000$ moves is j in the computed permutation. In the plots of the best position, a dot in the i, j entry denotes a probe in the ith position of the identity permutation, and whose position is j in the best computed permutation, *i.e.* that which minimizes the number of implied false negatives.

The plots we present are particular to the instances of the problem, but we have chosen plots which are representative of the algorithms, given particular parameters. The times are in seconds, and the programs were run on a SUN SPARC2.

Figures 6 and 10 show running times and plots for various combinations of the algorithms on the Chromosome 21 data.

Figures 7 and 11 through 14 show running times and plots for various combinations of the algorithms on simulated data with varying coverage.

Figures 8, 12, and 15 show running times and plots for various combinations of the algorithms on simulated data with varying false negative rate.

Figures 9 and 16 show running times and plots for various combinations of the algorithms on simulated data with a 30% false negative rate and no false positives or chimeric clones.

Sorting	Splitting	Screen/TSP/SA/Post	Screen/Sort/SA/Post	TSP	SA
1.5	14.6	45.3	44.9	17.9	331.0

Figure 6: Running Times on Chromosome 21 Data

Coverage	Sorting	Splitting	Screen/TSP/SA/Post	Screen/Sort/SA/Post	TSP	SA
10	2.9	4.8	79.4	376.5	15.3	1542.7
7	2.3	3.6	50.2	124.2	16.7	1250.1
6	2.1	3.2	47.5	48.0	22.0	1353.1
5	1.9	2.8	50.0	39.5	16.0	1281.4

Simulated Data: 200 clones, probe rate 1, False negative rate 20%, false positive rate 0.1%, chimeric clone rate 25%.

Figure 7: Running Times with Varying Coverage

8 Conclusions and Future Work

This paper has presented several algorithms for use in chromosome physical mapping with sequence tagged sites. In the absence of experimental errors, this is a simple task, but when errors are present, the problem is NP-hard. The Hamming distance TSP approach, and the maximum posterior probability simulated annealing approach have proved particularly robust in the presence of false positive, false negative, and chimeric errors. We have also been successful in screening out false positives and splitting chimeric clones.

There are several ways in which this work might be usefully extended. The first deals with incorporating other sources of information, the second deals with additional forms of errors, and the final extension addresses the direct analysis of pooled data.

Genetic linkage data, *in situ* hybridization data, and other information that provides the approximate location of STSs is often available, and we would like to have an integrated way of utilizing this knowledge. Furthermore, the PCR data may yield varying degrees of evidence that a probe occurs on a clone. Therefore a real-number representation may be preferable to a zero-one format. It is also important to accommodate missing data.

Secondly, our methods assume an error model that is incomplete. In particular, repeated STS sequences appear to be a problem in some data sets, such as the chromosome Y data. We anticipate that utilizing these other forms of information and explicitly recognizing a wider variety of errors will improve performance on many data sets.

The final extension that we are studying uses our programs to order STSs directly from pooled data. The standard approach to analyzing pooled data is to first disambiguate the STS-clone incidences, and then, using the disambiguated data, to order the STSs. In contrast to this, our

False Negative Rate	Sort	Split	Screen/TSP/SA/Post	Screen/Sort/SA/Post	TSP	SA
20%	2.3	3.6	50.2	124.2	16.7	1250.1
30%	2.3	3.6	67.3	84.4	22.5	1358.2

Simulated Data: 200 clones, probe rate 1, coverage 7, false positive rate 0.1%, chimeric clone rate 25%.

Figure 8: Running Times with Varying false negative rate

Sort	Split	Screen/TSP/SA/Post	Screen/Sort/SA/Post	TSP	SA
1.8	17.7	668.2	681.5	19.8	956.9

Simulated Data: 200 clones, probe rate 1, coverage 7, false positive rate 0%, false negative rate 30%, chimeric clone rate 0%.

Figure 9: Running Times with 30% false negatives, no false positives, and no chimeric clones

preliminary results indicate that the Hamming-distance TSP approach can produce good orderings operating directly on the pooled data. Essentially, each pool is viewed as a massively chimeric clone, and the robustness of the Hamming-distance TSP to this type of error is exploited.

Acknowledgement

We would like to thank Donn Davy of Lawrence Berkeley Laboratories for providing us with the data from human chromosome 21. We also thank Suzanna Lewis, Lee Newberg, Rudiger Reischuk and Ron Shamir for many fruitful discussions.

References

[ACT93] J. Arnold A.J. Cuticchia and W.E. Timberlake. ODS: Ordering DNA Sequences - a Physical Mapping Algorithm Based on Simulated Annealing. *CABIOS*, 9(2):215–219, 1993.

[AGL94] Richard Mott Andrei Grigoriev and Hans Lehrach. An Algorithm to Detect Chimeric Clones and Random Noise in Genomic Mapping. *Genomics*, 22:482–486, 1994.

[AKNW91] Farid Alizadeh, Richard M. Karp, Lee A. Newberg, and Deborah K. Weisser. Physical Mapping of Chromosomes: A Combinatorial Problem in Molecular Biology. In *Proc. 4th ACM-SIAM Symposium on Discrete Algorithms*, 1991. (to appear in *Algorithmica*).

[AL87] E. H. L. Aarts and P. J. M. Van Laarhoven. *Simulated Annealing: Theory and Applications.* 1987. D. Reidel Publishing Company.

[BL76] Kellogg S. Booth and George S. Lueker. Testing for the Consecutive Ones Property, Interval Graphs, and Graph Planarity Using PQ–Tree Algorithms. *J. Comp. and Syst. Sci.*, 13:335–379, 1976.

[B75] K.S. Booth. *PQ Tree Algorithms.* PhD thesis, University of California, Berkeley, 1975.

[CRG+92] Ilya Chumakov, Philippe Rigault, Sophie Guillou, Pierre Ougen, Alain Billaut, Ghislaine Guasconi, Patricia Gervy, Isabelle LeGall, Pascal Soularue, Laurent Grinas, Lydie Bougueleret, Christine Bellanné–Chantelot, Bruno Lacroix, Emmanuel Barillot, Philippe Gesnouin, Stuart Pook, Guy Vaysseix, Gerard Frelat, Annette Schmitz, Jean-Luc Sambucy, Assumpcio Bosch, Xavier Estivill, Jean Weissenbach, Alain Vignal, Harold Reithman, David Cox, David Patterson, Katheleen Gardiner, Masahira Hattori, Yoshiyuki Sakaki, Hitoshi Ichikawa, Misao Ohki, Denis Le Paslier, Roland Heilig, Stylianos Antonarakis, and Daniel Cohen. Continuum of overlapping clones spanning the entire human chromosome 21q. *Nature*, 359:380–386, October 1992.

[FVHP92] Simon Foote, Douglas Vollrath, Adrienne Hilton, and David Page. The Human Y Chromosome: Overlapping DNA Clones Spanning the Euchromatic Region. *Science*, pages 60–66, October 1992.

[GI94] David Greenberg and Sorin Istrail. The Chimeric Mapping Problem: Algorithmic Strategies and Performance Evaluation on Synthetic Genomic Data. *Computers and Chemistry*, 18(3):207–230, 1994.

[L88] P. Van Laarhoven. *Theoretical and Computational Aspects of Simulated Annealing.* 1988. Centrum voor Wiskunde en Informatica, Amsterdam.

[MBC91] Kenneth Lange Michael Boehnke and David R. Cox. Statistical Methods for Multipoint Radiation Hybrid Mapping. *American Journal of Human Genetics*, 49:1174–1188, 1991.

[MCG+93] Toru Mizukami, William I. Chang, Igor Garkavtsev, Nancy Kaplan, Diane Lombardi, Tomohiro Matsumoto, Osami Niwa, Asako Kounosu, Mitsuhiro Yanagida, Thomas G. Marr, and David Beach. A 13kb Resolution Cosmid Map of the 14 Mb Fission Yeast Genome by Nonrandom Sequence–Tagged Site Mapping. *Cell*, 73:121–132, 1993.

[PSM+91] Michael J. Palazzolo, Stanley A. Sawyer, Christopher H. Martin, David A. Smoller, and Daniel L. Hartl. Optimized strategies for sequence–tagged–site selection in genome mapping. *Proc. Natl. Acad. Sci. USA*, 88:8034–8038, 1991.

[RML93] Elmar Maier Jorg Hoheisel Richard Mott, Andrei Grigoriev and Hans Lehrach. Algorithms and Software Tools for Ordering Clone Libraries: Application to the Mapping of the Genome of Schizosaccharomyces Pombes. *Nucleic Acids Research*, 21(8):1965–1974, 1993.

[VFH+92] Douglas Vollrath, Simon Foote, Adrienne Hilton, Laura G. Brown, Peggy Beer-Romero, Jonathan S. Bogan, and David C. Page. The Human Y Chromosome: A 43–Interval Map Based on Natuarally Occurring Deletions. *Science*, pages 52–59, October 1992.

[YWA94] James Griffith W.E. Timberlake Yuhong Wang, Rolf A. Prade and Jonathan Arnold. A Fast Random Cost Algorithm for Physical Mapping. *Proceedings of the National Academy of Sciences USA*, 91:11094–11098, 1994.

[Z92] Geoffrey Zweig. An Effective Tour Construction and Improvement Procedure for the Traveling Salesman Problem. To appear in *Operations Research*.

Figure 1: Chromosome 21

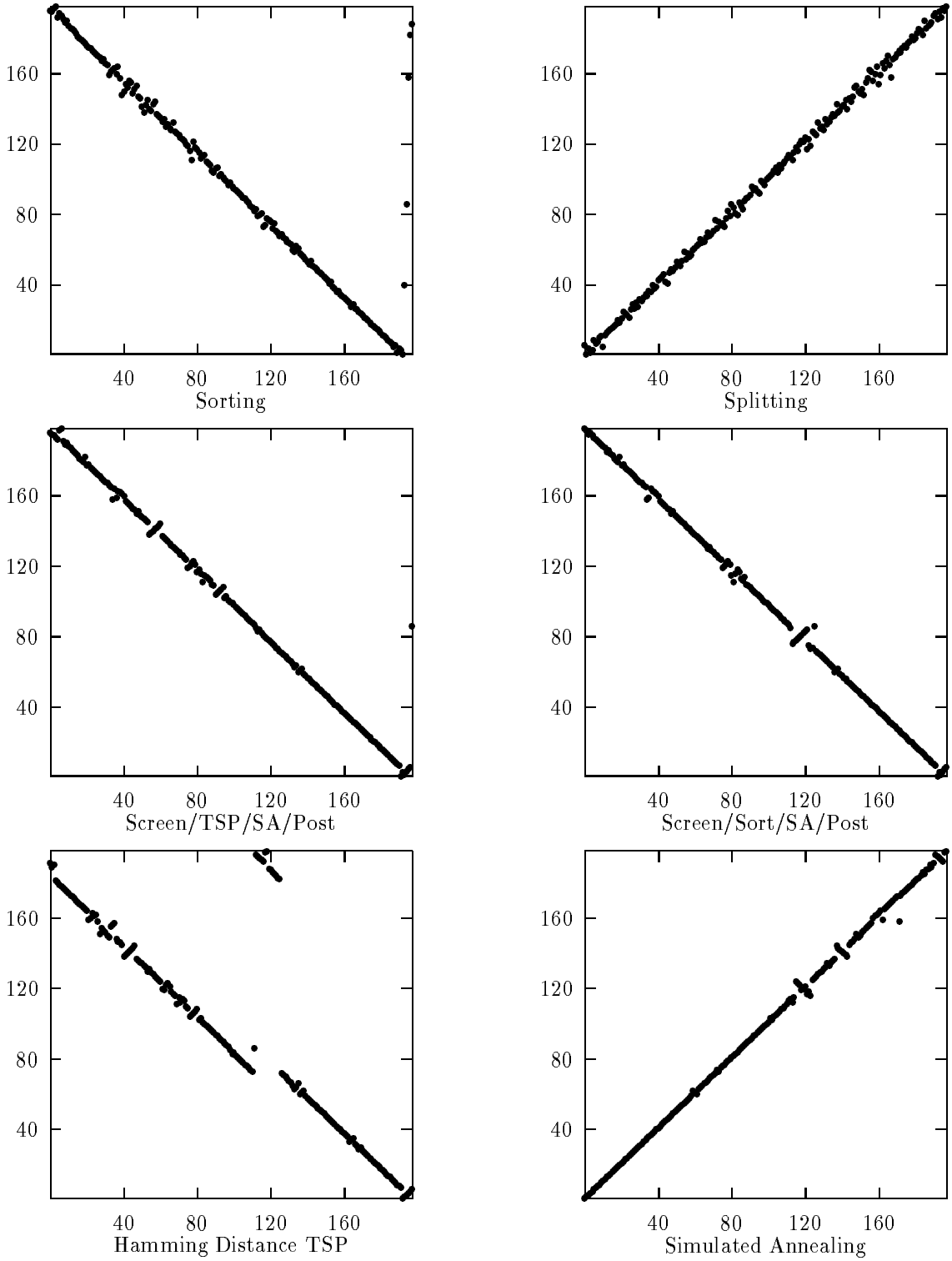

Figure 2: Coverage 10, 200 Clones, Probe Rate 1, False Negative Rate 20%, False Positive Rate .1%, Chimeric Clone Rate 25%

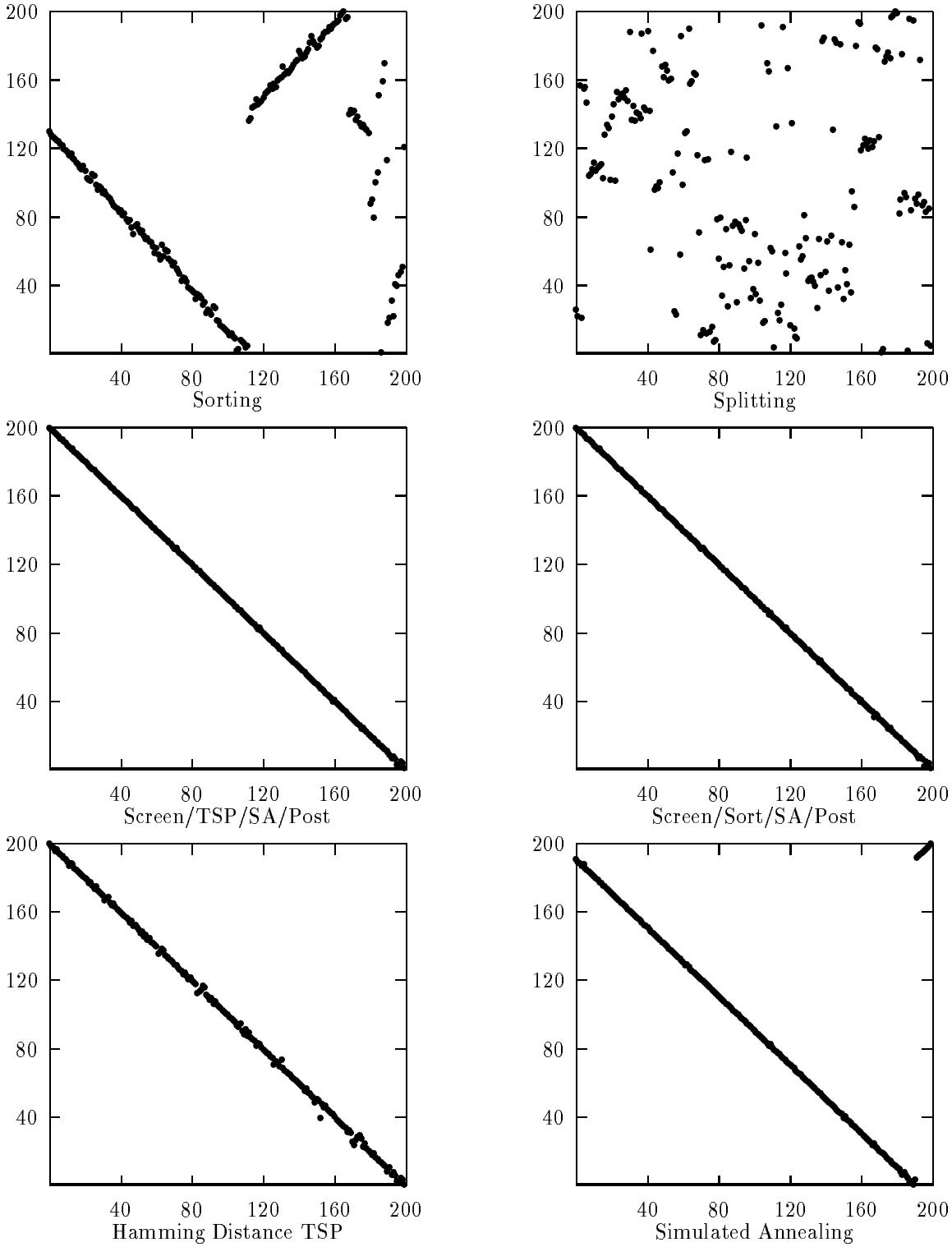

Figure 3: Coverage 7, 200 Clones, Probe Rate 1, False Negative Rate 20%, False Positive Rate .1%, Chimeric Clone Rate 25%

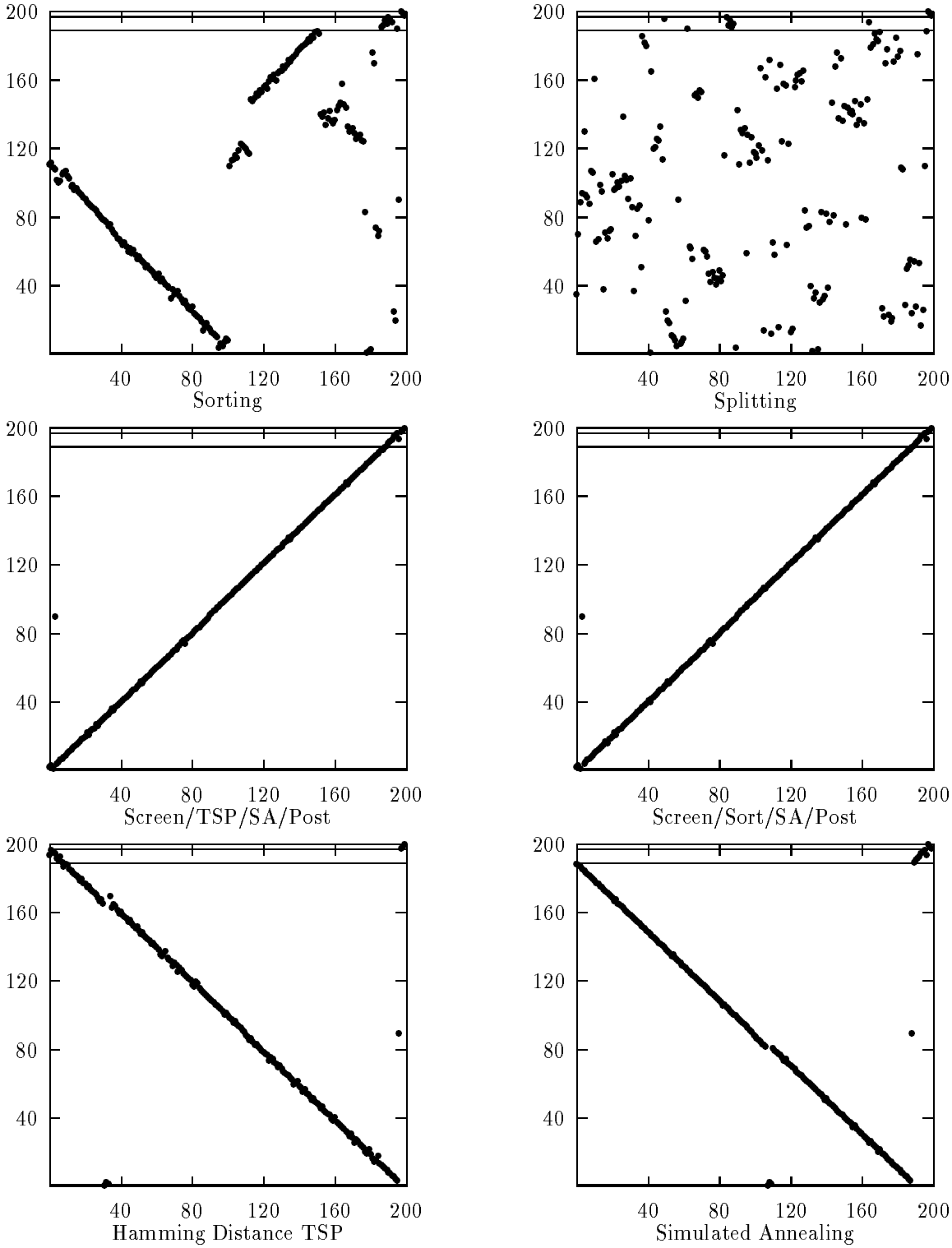

Figure 4: Coverage 6, 200 Clones, Probe Rate 1, False Negative Rate 20%, False Positive Rate .1%, Chimeric Clone Rate 25%

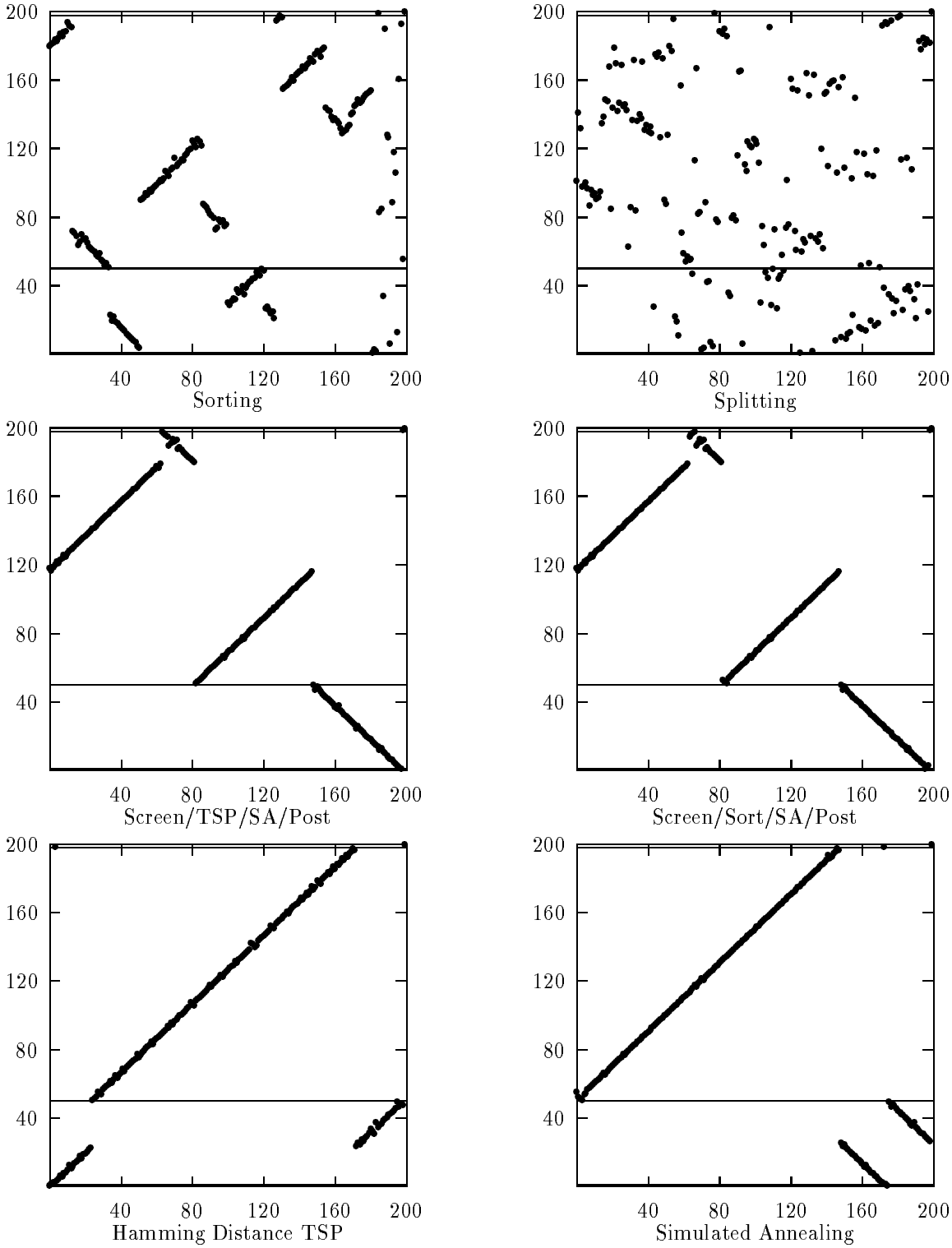

Figure 5: Coverage 5, 200 Clones, Probe Rate 1, False Negative Rate 20%, False Positive Rate .1%, Chimeric Clone Rate 25%

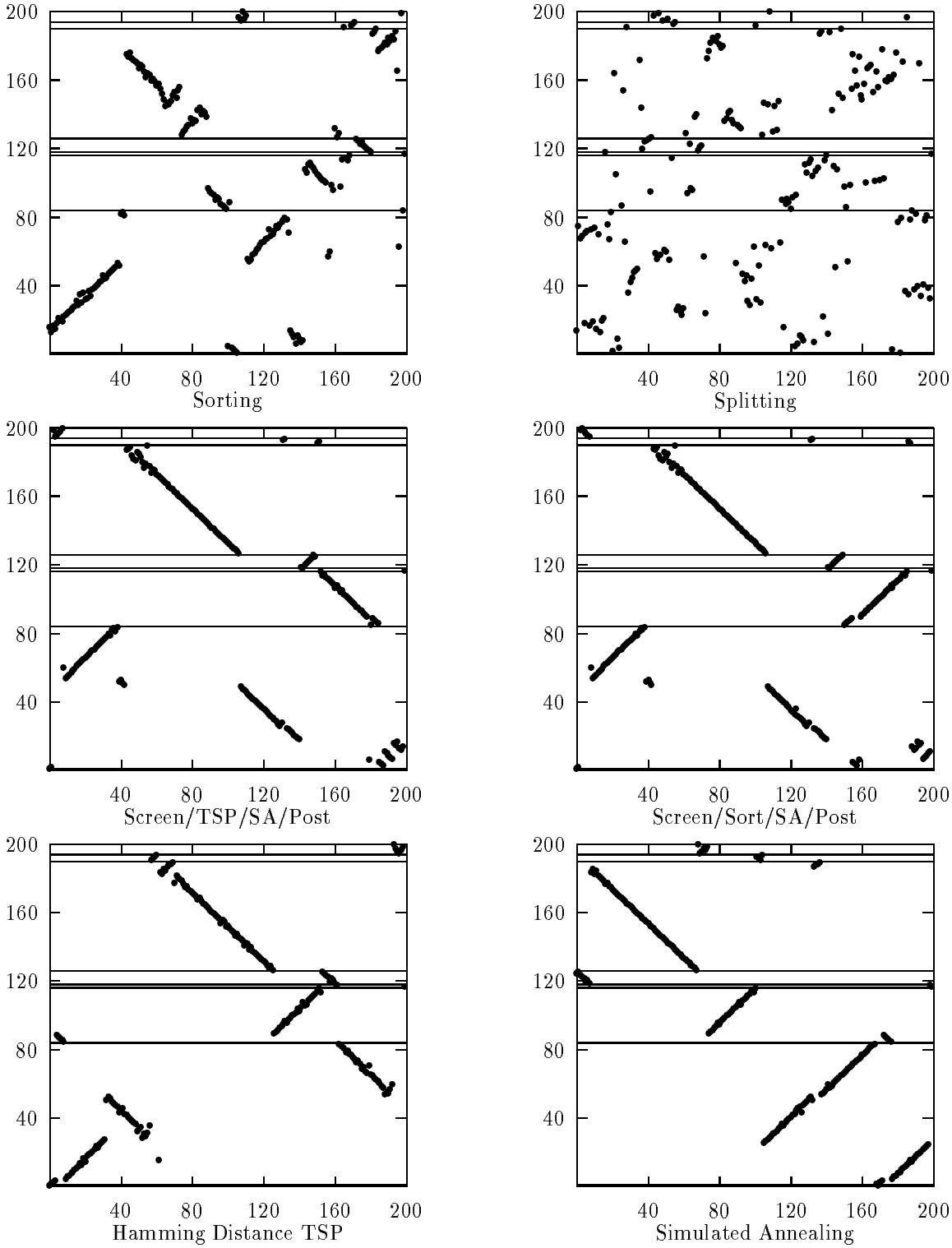

Figure 6: Coverage 7, 200 Clones, Probe Rate 1, False Negative Rate 30%, False Positive Rate .1%, Chimeric Clone Rate 25%

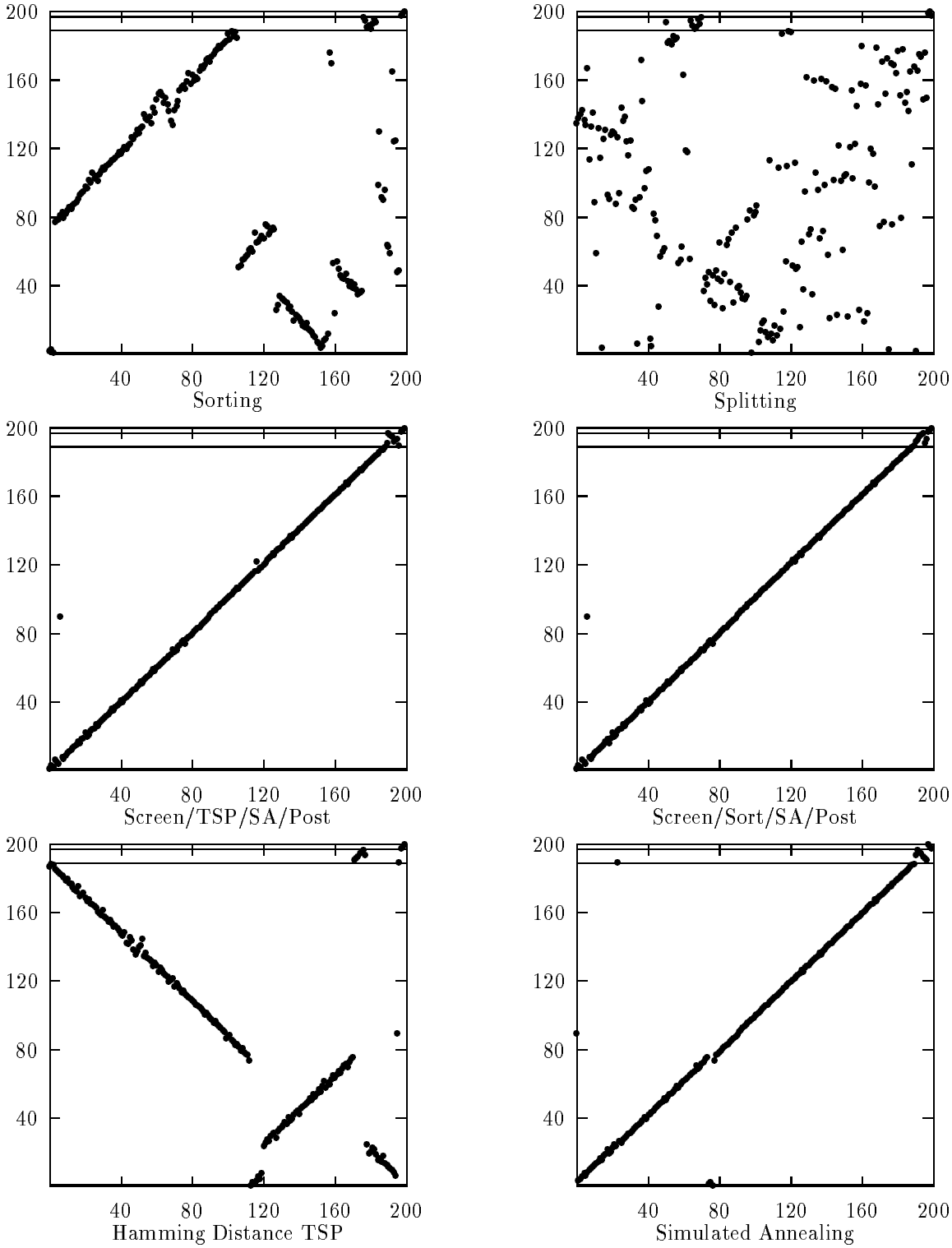

Figure 7: Coverage 7, 200 Clones, Probe Rate 1, False Negative Rate 30%, False Positive Rate 0%, Chimeric Clone Rate 0%

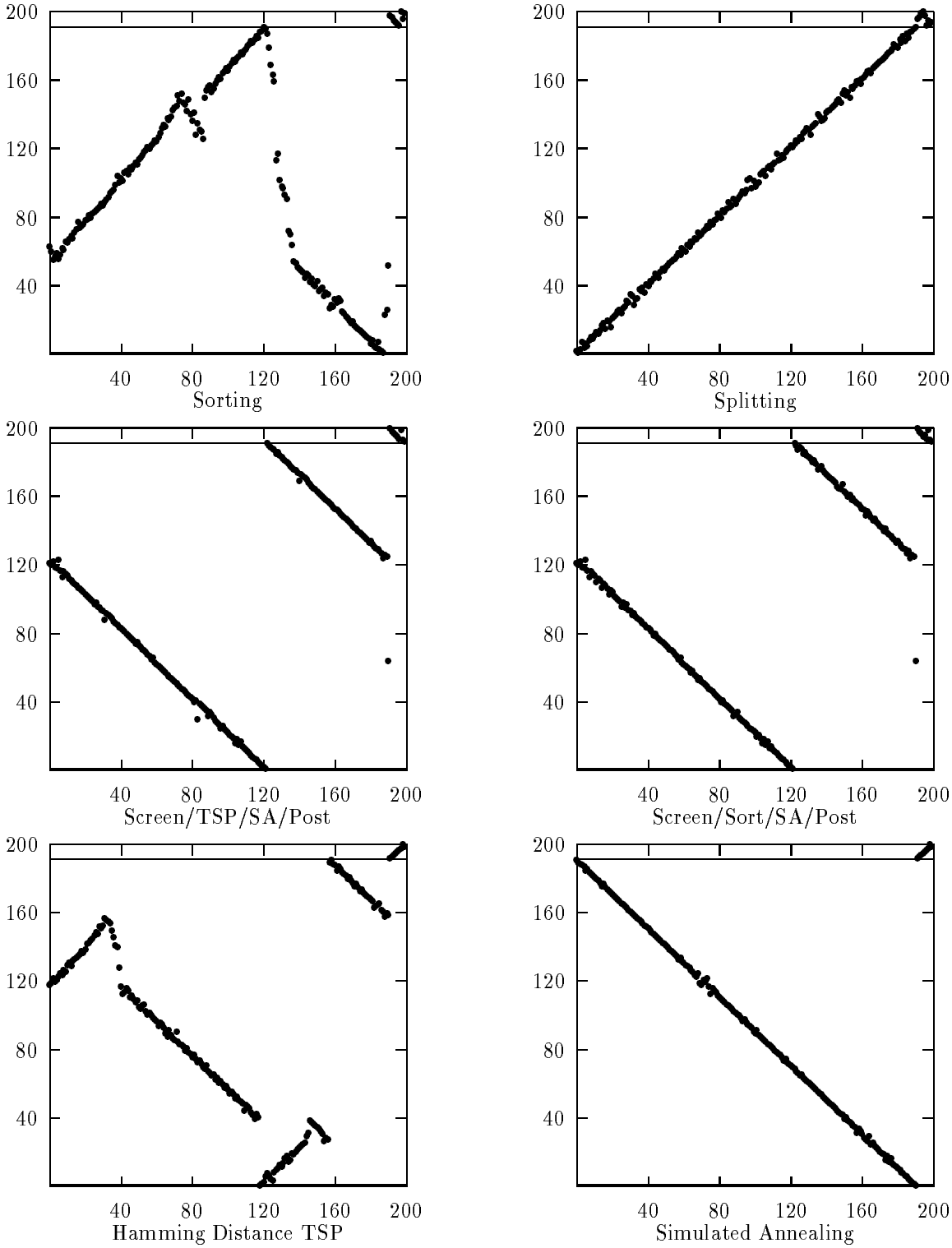

Figure 8: Average and best results of 40,000 random moves on Chromosome 21 data using Local Search neighborhood structure

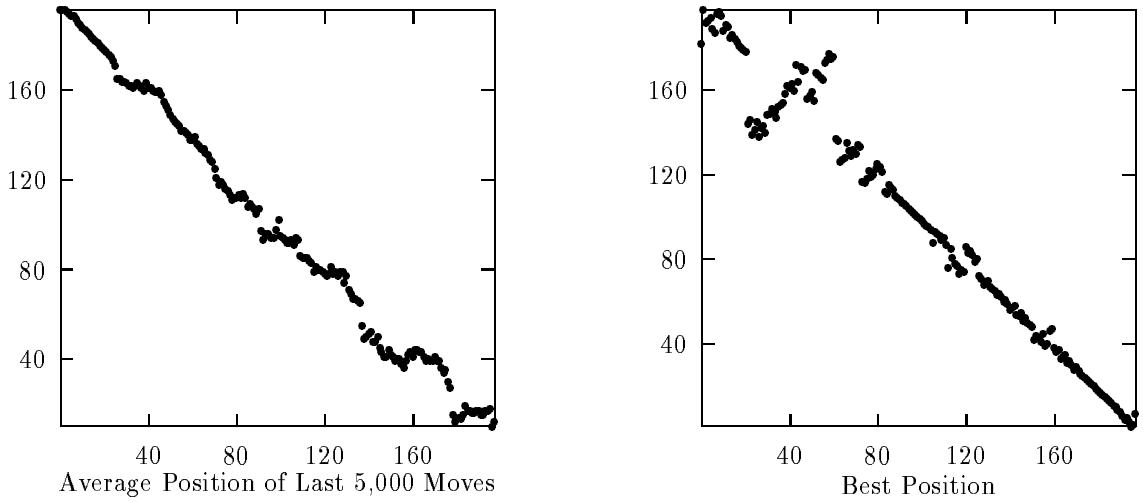

Figure 9: Coverage 7, 200 Clones, Probe Rate 1, False Negative Rate 20%, False Positive Rate 0%, Chimeric Clone Rate 0%