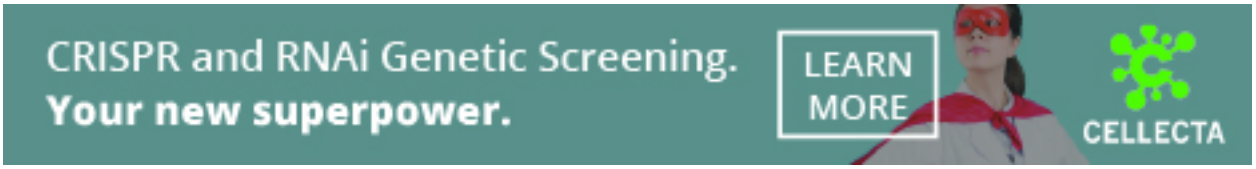
RHO––Radiation Hybrid Ordering

Amir Ben-Dor, Benny Chor and Dan Pelleg

Genome Res. 2000 10: 365-378
Access the most recent version at doi:10.1101/gr.10.3.365



RHO—Radiation Hybrid Ordering

Amir Ben-Dor,[1] Benny Chor, and Dan Pelleg[2]

Department of Computer Science, Technion, Haifa 32000, Israel

Radiation hybrid (RH) mapping is a somatic cell technique that is used for ordering markers along a chromosome and estimating the physical distances between them. With the advent of this mapping technique, analyzing the experimental data is becoming a challenging and demanding computational task. In this paper we present the software package RHO (radiation hybrid ordering). The package implements a number of heuristics that attempt to order genomic markers along a chromosome, given as input the results of an RH experiment. The heuristics are based on reducing an appropriate optimization problem to the traveling salesman problem (TSP). The reduced optimization problem is either the nonparametric obligate chromosome breaks (OCBs) or the parametric maximum likelihood estimation (MLE). We tested our package on both simulated and publicly available RH data. For synthetic RH data, the reconstructed markers' permutation is very close to the original permutation, even with fairly high error rates. For real data we used the framework markers' data from the Whitehead Institute maps. For most of the chromosomes (18 out of 23), there is a perfect agreement or nearly perfect agreement (reversal of chromosome arm or arms) between our maps and the Whitehead framework maps. For the remaining five chromosomes, our maps improve on the Whitehead framework maps with respect to both optimization criteria, having higher likelihood and fewer breakpoints. For three chromosomes, the results differ significantly (lod score >1.75), with chromosome 2 having the largest improvement (lod score 3.776).

We present a new software package, RHO, that gets as input the results of a radiation hybrid (RH) experiment and outputs an ordering of the genomic RH markers. RH mapping (Goss and Harris 1975, 1977a,b; Cox et al. 1990) is a somatic cell technique that is used for ordering markers along a chromosome and estimating the physical distances between them. It has several advantages over alternative mapping techniques, especially for intermediate scales' maps. The present biological techniques are capable of handling many hundreds (>1500) of markers in a single chromosome. With the advent of these techniques, analyzing the experimental data is becoming a challenging and demanding computational task.

The biological experiment in RH mapping has the following nature: Cells containing either one copy (haploid) or two copies (diploid) each of the chromosome of interest are irradiated. The radiation breaks the chromosomes at random locations into separate fragments. Each such fragment is a subinterval of the original chromosome, and it contains the markers within the subinterval. A random subset of the fragments is rescued by fusing the irradiated cells with normal rodent cells, a process that produces a collection of hybrid cells. The resulting clone may contain none, one, or many chromosome fragments. This recombinant clone is then tested for the presence or absence of each of the markers.

The algorithmic stage, which follows the biological experiment, aims at deducing the most likely linear order of the markers on the chromosome, based on the retention pattern of the markers in the different clones, and at estimating the spacing between markers. The intuition underlying this stage is the following: The further apart two markers are on the chromosome, the more likely it is that the radiation will create a break between them, placing the markers on two separate chromosomal fragments. Therefore, it is more probable that one of the markers will be present in the hybrid cell, whereas the other is absent. The retention pattern of any two markers can thus be used to estimate the physical distance between them.

Two different approaches are widely used to evaluate the quality of a proposed marker's permutation. The first is a combinatorial, nonparametric approach called obligate chromosome breaks (OCBs), whereas the second is the statistically based, parametric method of maximum likelihood estimation (MLE). The approach implemented in RHO is to translate either the OCB or the MLE formulation of the experimental results to the traveling salesman problem (TSP). The latter combinatorial optimization problem is then solved, and its solution is interpreted as a linear order of the RH markers, together with an estimate of their distances.

OCBs

The OCBs approach (Boehnke et al. 1991; Barrett 1992;

[1]Corresponding author. Present address: Department of Computer Science and Engineering, University of Washington, Seattle, Washington 98195 USA.
[2]Present address: Department of Computer Science, Carnegie Mellon University, Pittsburgh, Pennsylvania 15213 USA.
E-MAIL amirbd@cs.washington.edu; FAX (206) 543-2969.

10:365–378 ©2000 by Cold Spring Harbor Laboratory Press ISSN 1088-9051/00 $5.00; www.genome.org

Bishop and Crockford 1992; Boehnke 1992; Weeks et al. 1992; Lange et al. 1995) is analogous to minimizing the number of recombination events in the analysis of genetic linkage data (Thompson 1987). The goal is to find the markers' order under which the fewest number of radiation induced breaks must have occurred.

The number of OCBs implied by a RH is easily computed for any markers' order. An OCB is scored whenever a "+" (present marker) immediately follows a "−" (absent marker) or vice versa. In this scoring, "?" entries (ambiguous readings) are ignored. For example, the hybrid "+?+?−−??−−+" contributes two obligate breaks, one between the markers in positions 3 (a "+") and 5 (a "−"), the other between the markers in positions 9 (a "−") and 10 (a "+").

For a given order, the sum of OCBs over all hybrids is the overall score for the order. The objective function under the OCB criteria is to find the order that requires the fewest breaks. This order is suggested as the true markers' order.

MLE

In the MLE approach, the goal is to find a permutation of the markers and estimate the distance (breakage probability) between adjacent markers, such that the likelihood of the resulting map is maximized. The likelihood of a map (consisting of an ordered list of markers and the distances between them) is the probability to observe the RH data, given the map (in a suitable probability model of the experiment).

Markov model was used by Boehnke et al. (1991) and Lunetta and Boehnke (1994) to compute the likelihood of a map for haploid, error-free data. Lange et al. (1995) extended this approach to diploid data that accounts for laboratory errors as well, using a hidden Markov model. A very similar hidden Markov model is presented by Slonim (1996) and Slonim et al. (1996).

Given an order, estimating the breakage probabilities can be efficiently done in a provably optimal way for the Markov model (Boehnke et al. 1991) and heuristically for the hidden Markov models using the EM algorithm (Rabiner 1989; Slonim et al. 1996; Slonim 1996). Given the parameters of a specific panel, it is possible to convert breakage probabilities between adjacent markers to physical distances between them (Goss and Harris 1975; Newell et al. 1998). Different maps of the same chromosome thus give rise to different estimates of its total physical length. Shorter maps are generally viewed as more desirable ones as they usually correspond to maps with higher likelihood.

Software Packages in Use

An obvious barrier to optimizing for either OCB or MLE criteria is the shear number of orders that must be considered. For n markers, this number is $n/2$ (as an order and its reverse are equivalent), which is too large to search exhaustively, even for moderate values n. In the worst case, both criteria give rise to computationally intractable (NP-hard) optimization problems. Various search strategies have been proposed in the literature (Boehnke et al. 1991; Bishop and Crockford 1992; Boehnke 1992; Lange and Boehnke 1992; Lunetta and Boehnke 1994; Lange et al. 1995; Lunetta et al. 1995; Slonim 1996; Stein et al. 1997), and several software packages implementing these strategies are available. See http://linkage.rockefeller.edu/tara/rhmap for information on RH mapping. The first is RHMAP, written by Boehnke's group (Boehnke et al. 1996), which optimizes with respect to both the OCB criterion and to the MLE criterion for various models.

A different software package, RHMAPPER, was developed at the Whitehead Institute (Slonim et al. 1996). This software optimizes with respect to the MLE criterion, using a hidden Markov model that accounts for laboratory error (Slonim 1996; Stein et al. 1997). RHMAPPER first produces a sparse framework map, containing a subset of the markers whose relative order is deduced with high certainty. The construction of the framework map uses a heuristic to solve the betweenness problem (Opatrny 1979; Chor and Sudan 1998). The rest of the markers are placed into bins relative to the framework markers to form a placement map.

The package SAMMAPPER (Stewart et al. 1997) divides the markers to strongly linked groups and orders these groups. The optimization criteria is maximum likelihood. Simulated annealing is applied to search the likelihood space. Unlike RHMAPPER, this package does not use framework markers to build its map on. A different approach, based on information theoretic tools, is used by Newell et al. (1998). Distances between pairs of markers are estimated from the experiment outcome, using mutual information. These pairwise distances are then transformed into a one-dimensional, linear map by methods of distance geometry (Newell et al. 1995).

Recently, a number of linkage mapping software packages were extended to provide support for RH mapping: Map Manager developed by Manly and Olson (1999), Map+ developed at the University of Southampton, UK, and MultiMap/RADMAP, developed at the Rockefeller university (Matise et al. 1994).

An Overview of RHO

Our system is set to extract reliable markers ordering information from the RH data. Because, in practice, the number of typed markers is too large to be reliably ordered, a simple optional pruning step is first performed. In this step we choose a representative set of framework markers that will be later ordered to form a reliable framework map. As a final step, we can assign the rest (nonframework) of the markers into bin using

the framework map. This step is not supported in the current version or RHO.

As both the OCB and MLE optimization problems are computationally hard, we cannot hope to give an efficient algorithm that provably solves them in a worst case scenario. Our approach is to reduce both optimization problems to the TSP. This reduction (described in Methods) has two important advantages: (1) The technology for solving large TSPs is quite advanced, so near-optimal solutions are obtained fairly easily; and (2) the special structure of the TSP allows for efficient computation of tight lower bounds for the cost of the optimal solution. This can be done using the *Held–Karp* method (Held and Karp 1971).

We use two methods to solve the resulting TSP instances: The *Simulated Anneling* heuristic and the *Held–Karp* method. The *Simulated Annealing* heuristic (Press et al. 1992) with the *2-OPT* neighborhood structure (Lin and Kernighan 1973) is very efficient to compute and is known to be highly effective for TSP. We note that *Simulated Annealing* was already used for RH mapping (e.g., in RHMAP; also see Slonim 1996). However, the big difference is that previously, *Simulated Annealing* was applied directly to either OCBs or MLE. For those problems, computing the cost of moving to a neighbor permutation requires either scanning the entire data matrix (in the OCB case) or an expensive EM execution (in the MLE case). In our approach, using the TSP formulation, the transition can be computed in constant time. Thus, many more neighbor moves can be performed, increasing significantly the efficiency of the search.

The *Held–Karp* method (Held and Karp 1971) produces tight lower bounds on the cost of any TSP solution. The basic idea is to use the fact that a minimum spanning tree (MST) provides a lower bound on the cost of a traveling salesman tour. After constructing a tree, the weights of the graph are modified in a way that does not change the identity of the optimal tour but gives a "high penalty" to nodes with high degree. A new MST is found in the modified graph, and this process is iterated, producing better and better lower bounds. The tightness of the lower bounds can be estimated by comparison to the upper bound from the simulated annealing. We are not aware of any method to directly produce tight lower bounds for either the OCB or the MLE objective functions.

RESULTS

Synthetic RH Data

We first tested our algorithm for synthetic RH data. This data is produced by the simulator, whose "experimental parameters" are controlled by the user. The resulting RH data matrix D serves as the input for our ordering algorithm (with respect to both MLE and OCB criteria). In many of the simulations, our parameters are close to these of the Genebridge 4 RH panel. All data sets of the synthetic RH experiments correspond to a diploid RH panel, containing 93 hybrids, with retention rate 0.15 (per chromosome), and breakage rate 10. Moreover, as usually done in practice, we assumed that for each marker/hybrid pair, the typing was done twice. In the simulations, we varied the number of markers, n (between 132 and 400), and the reading error rates α (false positive) and β (false negative) (between 0.01 and 0.1). In addition, we tested our approach of choosing framework markers by varying the value of a threshold parameter θ_o between 0 (no pruning at all) and 0.2. If pruning was done, we report the average number of framework markers chosen, k. The precise set of parameters used in the seven data sets appears in Table 1.

For each data set, we have simulated the experiment 10 times. The resulting permutation was compared with the original permutation. For 132 markers (data sets 1 and 2), the resulting order was very close to the original order. Typical results are shown in Figure 1. In this figure we plot the original order of the markers along the x-axis. The y-coordinate is the position of the markers in our order. Notice that even for 400 markers (data set 3), the resulting permutation is close to the correct one (Fig. 2). In the rest of the experiments (data sets 4, 5, 6, and 7), we applied our pruning procedure to choose framework markers. We set the threshold parameters, θ_o, experimentally, such that ~100 markers will be chosen from the 400 markers. We ordered the resulting set of the framework markers and compared the output permutation to their true order. In all the tests for data set 4, the *Held–Karp* method found the TSP optimal order (which was the same one for both the OCB and MLE criteria). Moreover, this order is the exact original order of the markers. This outcome validates our pruning approach for realistic parameters.

To test the effect of higher error rates, we introduced data sets 5, 6, and 7. For data set 5 we had essentially the same excellent results as in data set 4. The difference between the true order and the order re-

Table 1. Parameter Sets Used to Produce Synthetic RH Data

	n	α	β	θ_o	k
1	132	0.03	0.03	0.0	
2	132	0.01	0.01	0.0	
3	400	0.01	0.01	0.0	
4	400	0.01	0.01	0.1	92
5	400	0.03	0.03	0.1	93
6	400	0.01	0.05	0.1	97
7	400	0.1	0.1	0.2	106

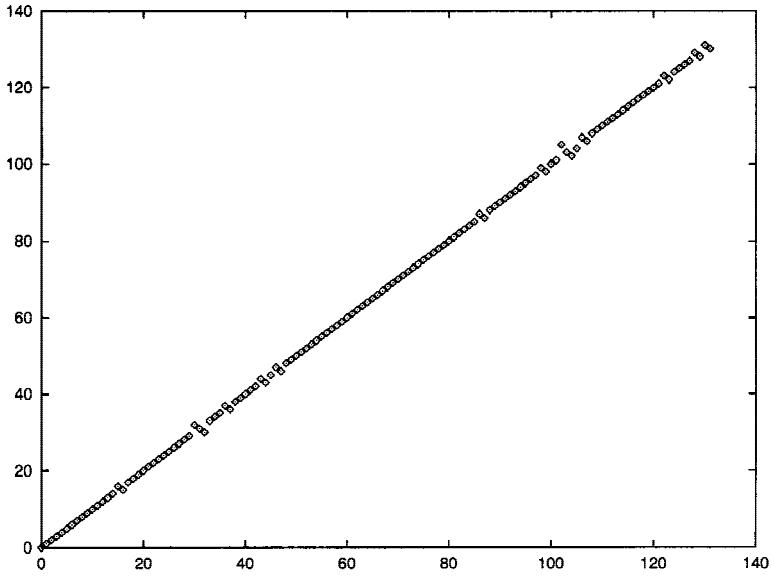


Figure 1 Order found for synthetic data set 1 (*y*-axis) vs. original permutation (*x*-axis).

turned by the algorithm was at most a single reversal of two adjacent markers.

In data set 6 we test the realistic scenario in which the false-positive rate is smaller than the false-negative rate ($\alpha = 0.01$, $\beta = 0.05$). In all 10 experiments for this data set, our algorithms produced the optimal TSP ordering (with respect to the corresponding objective function). In 5 out of 10 experiments the returned permutation was the original markers' permutation. In the other five cases the original permutation was not TSP optimal, because of "typing errors." However, in these cases the differences between the original and the reconstructed (optimal) permutation consisted of a small number of reversals between pairs of adjacent markers. In the worst case (which occurred in 1 out of 10 experiments) exactly three such reversals occurred.

In data set 7, we considered very high rates of typing errors (10%). In this case, the framework maps are close but not identical to the true order (e.g., see Fig. 3). However, there is a noticeable degradation in their visual closeness. This gives an indication that such error rates might be problematic for our heuristics. We note that such high error rates are probably overly pessimistic for RH experiments.

Real RH Data

The RH data used to construct the maps was downloaded from the Whitehead Institute for Biomedical Research. (http://www.genome.wi.mit.edu/ ftp/distribution/human_STS_releases/july97/rhmap/.). This data was gathered as follows: The RH panel used was the Genebridge 4 RH panel (Walter et al. 1994), containing 93 hybrids, constructed from a diploid donor cell line. The screening was performed at the Whitehead Institute. For each DNA marker, the screening results are reported as a vector of 93 0's, 1's, and 2's. Each entry corresponds to 1 of 93 cell lines in the radiation hybrid panel. A "0" and a "1" represent negative and positive readings (PCR assays), respectively. A "2" indicates that the assay gave contradictory results for two or more replicated typings.

Even though the RH data contain the retention patterns for all markers, we have restricted our attention to those markers selected as framework markers by the Whitehead team (using RHMAPPER), as the other markers (called placement markers) are only assigned approximate locations. The number of framework markers, as well as the total number of markers in each chromosome, appears in Table 2.

For each chromosome and each optimization criteria (OCB and MLE) we applied our algorithms (*Simulated Annealing* and *Held-Karp*) to search for the best ordering with respect to the criteria. Because the true permutation is unknown for real data, we compared our proposed ordering for each chromosome to the order in the Whitehead Institute (WI) maps (final data release, July 1997, of the WI). (Hudson et al. 1995).

For each of the two competing permutations, we

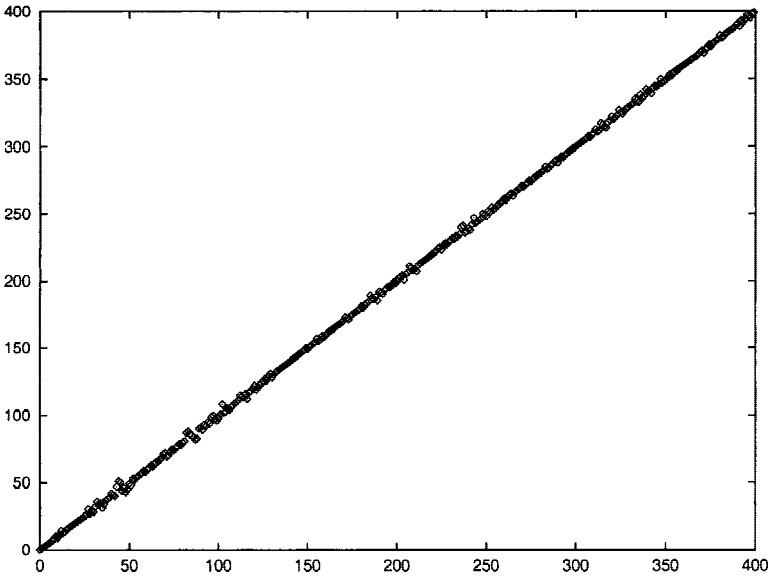


Figure 2 Order found for data set 3 (*y*-axis) vs. original permutation (*x*-axis).

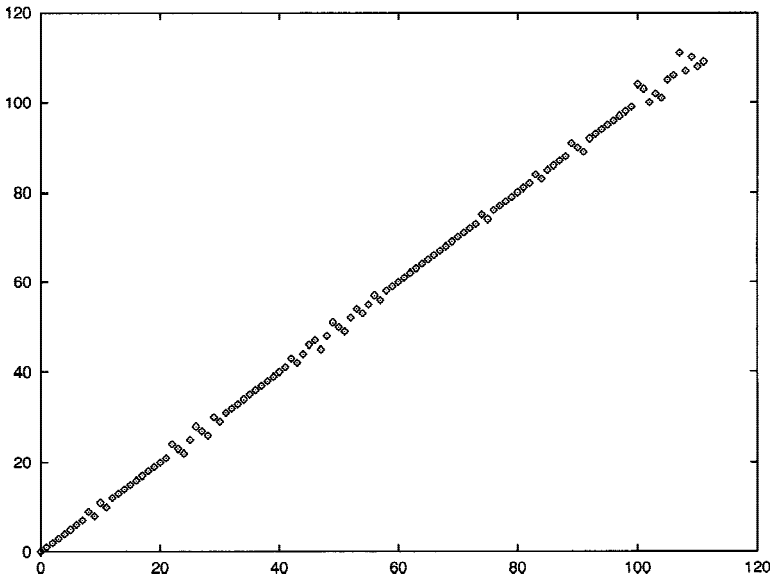


Figure 3 Order found for framework markers in synthetic data set 7 (*y*-axis) vs. true permutation (*x*-axis).

attach two scores: its likelihood (MLE score) and the number of OCBs it induces (OCB score). The number of breaks is computed directly (see introductory section). To compute the likelihood of a permutation, we used the EM algorithm as implemented in RHMAPPER (Slonim et al. 1996). This procedure evaluates the most likely breakage probabilities between adjacent markers (in a hidden Markov model) and the resulting likelihood of the order. For this computation we used a retention rate of 0.15 and error rates and missing data rates of 0.01, as was used in the construction of the WI maps.

The difference between the OCB score and the ratio of likelihood of two different permutations provides a quantitative measure of comparison between the two. Because the likelihood of any particular map is extremely small, we measure the difference between the (base 10) logarithms of the likelihood instead of direct likelihood ratio. This difference is the lod score of the two orders.

For 14 out of 23 chromosomes, there is a complete agreement between the current framework map and our framework map (chromosomes 1, 4, 7, 10, 11, 12, 13, 14, 15, 18, 19, 21, 22, and X). For four out of the remaining nine chromosomes (9, 16, 17, and 20), our permutations consist of a complete reversal of chromosome arm or arms. For these chromosomes the suggested arm reversals typically either induce fewer breakpoints or are more likely than the present permutations.

At first sight it may seem that two maps with reversed arms are very different. However, there is a simple explanation for this discrepancy. It is known that in the WI data there is very low linkage between markers on the opposite sides of the centromere (Slonim 1996). That is, markers in different sides of the centromere are retained independently in almost all hybrids. Therefore, RH mapping is not very well suited to orient the chromosomes' arms but, rather, to order markers within each arm. We conclude that for 18 out of the 23 chromosomes, our maps are in fact the same as the WI maps.

For the remaining five chromosomes (chromosomes 2, 3, 5, 6 and 8), our maps are more likely to be correct and induce fewer breakpoints. For three out of these five chromosomes, the TSP instance were solved optimally (using the *Held–Karp* algorithm). For the other two chromosomes (2 and 6), there is a small gap between the lower bound (*Held–Karp*) and the upper bounds (*Simulated Annealing*). The differences are 0.7% for chromosome 2, and 1.1% for chromosome 6.

The results, described in Table 3, are sorted according to the lod score of the new map (compared with the WI map). The third column in the table lists the difference between the OCB counts. For example, our permutation for chromosome 2 has a lod score of 3.776 compared with the current permutation (i.e, it is ~6000 times more likely than the WI map). Moreover, the new permutation induces four fewer OCBs than the current one. Finally, the fourth column describes the optimization criteria (MLE or OCB) used to obtain the permutation.

Table 2. Total Number of Markers and Framework Markers in Each Chromosome

Chromosome	Total no. of markers	Framework markers
1	1379	132
2	1182	132
3	1060	113
4	761	105
5	814	75
6	833	112
7	808	74
8	634	86
9	637	76
10	701	91
11	826	75
12	761	75
13	364	50
14	513	43
15	505	49
16	446	37
17	460	43
18	335	55
19	378	30
20	356	40
21	145	25
22	273	21
X	494	54

Table 3. Comparison Between the WI Maps and our Maps

Chromosome	lod score	OCB delta	Optimization criteria
2	3.7662	4	MLE
8	2.0691	0	MLE
5	1.7525	2	OCB
3	1.0047	2	MLE
6	0.5544	2	MLE

The framework maps of chromosomes 3, 5, 6, and 8 are shown in Figure 4. Each graph plots the current (WI) order of the markers along the *x*-axis. The *y*-coordinate is the position of the markers in our order.

A New Map of Chromosome 2

We now describe the details of our chromosome 2 map and compare it with the WI map. The names of the 132 framework DNA markers, ordered according to the WI map, appear in Table 4. Using the RH data for these 132 framework markers, a TSP instance was produced for the MLE optimization problem. Then, the markers were ordered using the `Simulated Annealing` implementation. This permutation is presented in Figure 5. The graph plots the current (WI) order of the markers along the *x*-axis. The *y*-axis shows the same markers in their position according to the new order. Although the two maps are quite similar, there are four differences between the new and the current order:

1. The order of the markers WI-8825, WI-6849, WI-4861, D2S373, CHLC.GATA88C05, and D2S176 is reversed in the new map (first reversal).
2. The order of the markers D2S121, CHLC.GAAT11C03, AFMA037YB9, AFM151XD12, WI-6701, AFM120XC11, CHLC.GATA3D02, AFM309XB1, and WI-3192 is reversed in the new map (second reversal).
3. The marker D2S110 appears after the markers NIB1668 and WI-3307 in the new map.
4. The markers WI-2535, WI-5538, WI-5555, D2S324, WI-6410, and WI-4121 are reversed in the new map (third reversal).

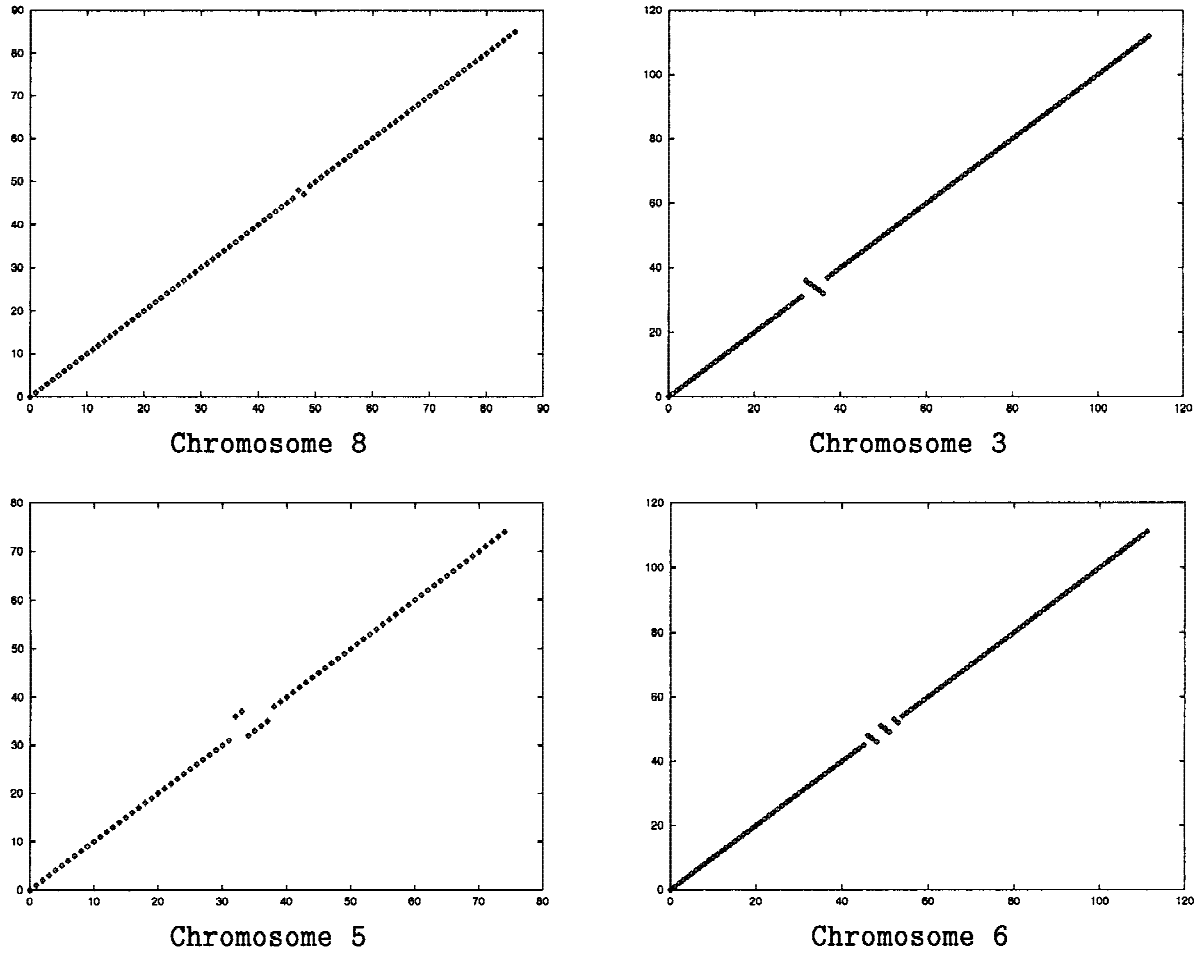


Figure 4 Our framework maps (on *y*-axis) vs. the WI map (on *x*-axis).

Table 4. The Framework Markers of Chromosome 2

Marker	Marker	Marker	Marker
AFMA070WC9	CHLC.GATA23H01	CHLC.GATA3D02	WI-6410
WI-1412	D2S357	AFM309XB1	WI-4121
D2S323	WI-6241	WI-3192	CHLC.GATA85E09
D2S330	D2S337	D2S110	IB2065
WI-6565	WI-3580	NIB1668	AFMB353YE9
D2S359	D2S290	WI-3307	WI-3652
AFMA323ZH1	CHLC.GATA72A05	AFM200VG7	D2S117
WI-5477	WI-10237	WI-5343	AFMA121YH1
CHLC.GATA23G07	D2S292	WI-3726	D2S348
WI-5509	WI-10809	CHLC.GATA27A12	WI-10756
WI-3767	CHLC.GATA69E12	AFMA153WB1	WI-5293
D2S387	WI-5987	AFMA338YE5	WI-3694
GCT11G10	GCT1B4	CHLC.GATA8H05.481	AFMA337XG9
CHLC.ATC5A09	D2S139	AFMA067ZE1	WI-1247
WI-5693	CHLC.GATA26G08	WI-4819	AFMB299WB5
AFMB353WF1	CHLC.GATA71D09	WI-3759	D2S317
WI-3742	AFMB297YC5	WI-4650	WI-4668
WI-5262	AFMA067XH5	WI-5772	WI-6439
D2S174	CHLC.GATA88G05	D2S151	D2S295
WI-4431	WI-6616	WI-4798	CHLC.GATA3F05
CHLC.GATA8F07.440	CHLC.GATA85A06	D2S142	D2S339
D2S352	WI-8825	WI-4058	AFM348TG5
WI-9798	WI-6849	WI-6698	D2S360
D2S177	WI-4861	CHLC.GATA22H09	CHLC.GATA2E04.760
WI-10326	D2S373	WI-6270	D2S353
WI-3893	CHLC.GATA88C05	D2S354	CHLC.GATA52F12
AFM210XE9P	D2S176	D2S111	D2S159
WI-3220	D2S121	CHLC.GATA71B02	D2S396
WI-4108	CHLC.GAAT11C03	WI-3728	WI-9093
AFM196XF6	AFMA037YB9	WI-2535	D2S331
WI-3027	AFM151XD12	WI-5538	WI-6310
WI-4077	WI-6701	WI-5555	D2S2986
WI-6613	AFM120XC11	D2S324	D2S2585

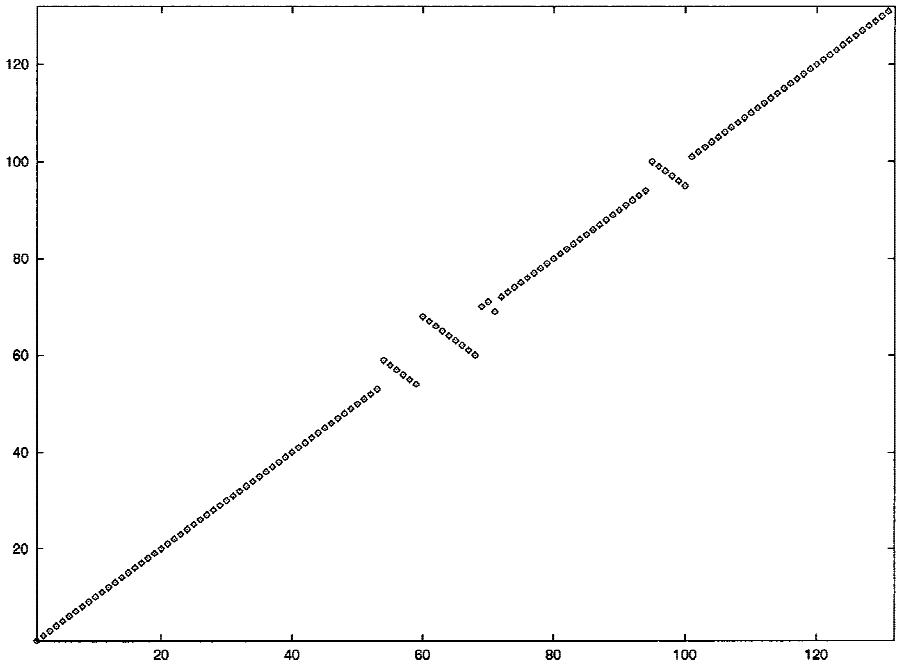


Figure 5 Our framework map for chromosome 2 (*y*-axis) vs. the WI framework map (*x*-axis).

Given the parameters of a specific panel, it is possible to convert breakage probabilities between adjacent markers to physical distances between them (Goss and Harris 1975; Newell et al. 1998). Different maps of the same chromosome thus give rise to different estimates of its total physical length. Shorter maps are generally viewed as more desirable ones. This transformation of probabilities to distances is implemented in RHMAPPER. Using this implementation, we conclude that the total physical length of chromosome 2 in our map is 3.88% shorter than in the WI framework map. (Our map length is 1582.4 CR, whereas the WI map length is 1646.2 cR.) The detailed differ-

ences between the two maps are depicted graphically in Figure 6. These map portions are drawn to scale. The numbers on the left hand side of each figure denote distances (in centiRay) on the chromosome. If there is enough space on the figure, distances between adjacent markers are also printed.

Availability and Performance

The RHO software is freely accessible on the Web (Ben-Dor and Pelleg 1998) as a CGI script. The user can supply his or her own data files in the WI format (Hui 1997) (with up to 400 markers) or experiment with existing files from the Whitehead distribution of July 1997. All program parameters are adjustable via user-friendly menus presented on the browser window. The entire process is run on our servers (i.e., there is no need to download software). Once the processing is complete, usually within a few minutes of the request, an e-mail message is sent to the user. The user's browser can then be used to view the results both graphically and numerically. To access our site, go to http://www.cs.technion.ac.il/Labs/cbl/research.html and click on "RHO." Typical running times (measured on a Sun Ultra-4 at 300 MHz) for the Whitehead RH data on chromosomes 2 and 3 (132 and 113 framework markers, respectively) are as follows: *Simulated Annealing* is executed 30 times (each time from a random initial permutation), and the best result is returned. These 30 executions together take ~100 sec. The *Held–Karp* algorithm is executed once. It takes between 5 sec (for the graphs induced by MLE distances) and 18 sec (for the graphs induced by OCB distances).

DISCUSSION

Our goal in designing RHO was to implement heuristics, applicable to realistic data sets, that produce correct RH maps efficiently. Our approach is based on reducing both the OCB and the MLE optimization problems to the TSP. This reduction has two important advantages. First, it allows us to apply the *Simulated Annealing* heuristic that is known to be very efficient for TSP instances. Second, the structure of the TSP allows us to obtain the tight lower bound of the optimal solution (using the *Held–Karp* algorithm). Quite unexpectedly, the *Held–Karp* algorithm also turned out to be applicable for ordering large subsets of the markers. In all maps, the longest path in the *Held–Karp* tree contained >65% of all markers (and >95% for all 23 WI maps). Moreover, each one of the other markers (those

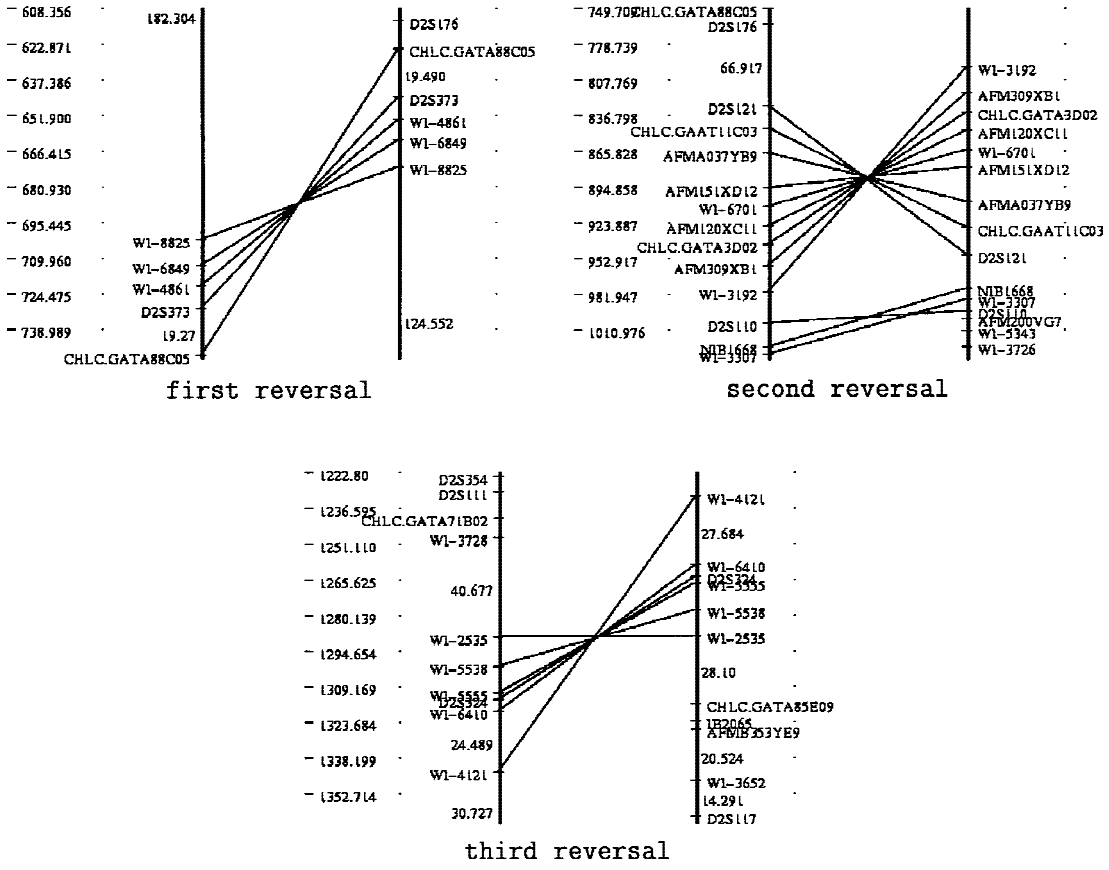


Figure 6 A close look at the differences between our chromosome 2 map (*right*) and the WI framework map (*left*).

that do not belong to the longest path) forms a branch of length one from the longest path. Thus, the markers along the longest path provide a good choice for a partial ordering of the markers.

We have implemented these algorithms and tested them on synthetic data (where we know the correct answer), produced according to the statistical model of the experiment, using realistic parameters. The returned order (for both OCBs and MLE) was very close to the true order (with up to 400 markers). Applying our pruning method to the data, and ordering the resulting subset of "framework markers" (~100 markers), we got excellent results—in most cases the original permutation was returned using both optimization criteria (the exception occurred when error rates were as high as 10%). Thus, we feel confident that our heuristics are applicable to RH mapping.

A comparison between the OCB and MLE scores of the returned permutations and the original permutations revealed an interesting phenomenon: In all cases where the output permutation was different from the original permutation, it turned out that the outputs of the algorithms are "better" than the original in terms of our objective functions. This phenomenon suggests that using only a moderate number of hybrids (as is done in practice), both objective functions are adequate to order only a few hundred markers. Above this number, the original permutation is typically not the optimal one, and thus we might mistakenly reject the true order in favor of a permutation with a better score.

We have applied our algorithms to real RH data and compared the resulting order (again, with respect to both optimization criteria) to the WI framework maps (Hudson et al. 1995). The results are very encouraging. For 14 out of the 23 chromosomes (chromosomes 1, 4, 7, 10, 11, 12, 13, 14, 15, 18, 19, 21, 22, and X), there is a perfect agreement between our maps and the WI framework maps. For four out of the remaining nine chromosomes (chromosomes 9, 16, 17, and 20), our permutations consist of a complete reversal of chromosome arm or arms. This means that for 18 out of the 23 chromosomes, our maps are essentially the same as the WI maps. We feel that this agreement gives substantial additional support to the validity of our heuristics. This also supports the correctness of these maps, as the same orders were derived by very different methods.

For the remaining five chromosomes (chromosomes 2, 3, 5, 6, 8), our maps improve on the WI framework maps with respect to both optimization criteria. They have higher likelihood and induce fewer breakpoints. Although for some chromosomes the differences are fairly small, there are three chromosomes (chromosomes 2, 8, and 5) where the results are noticeably better (lod score >1.75). In particular, for chromosome 2, our order is 6000 times more likely to be correct (lod score 3.776), and our map is 3.88% shorter than the WI map. We note that some of the markers that are mapped in the controverisal intervals have been also typed with the G3 panel by the Stanford Genome Center and were ordered by SAMMAPPER. In particular, the Stanford map seems to be in agreement with the WI map regarding the first reversal (based on markers DS2373, CHLC.GATA88C05, and D2S176). On the other hand, for the second interval, the Stanford map is in agreement with our map (based on markers DS121 and CHLC.GAAT11C03).

Because our algorithms differ substantially from other algorithms currently in use, we think they form a natural choice for an "independent verification" of RH mapping results. Cases where our results differ from reported results by one or two reversals in the framework markers' permutation can be resolved fairly easily by either additional small scale RH experiments (involving only a few markers) or by alternative techniques (e.g., FISH; Johnson et al. 1991).

METHODS

The Simulator

To test the effect of various experiment parameters (e.g., number of markers vs. number of hybrids and radiation level vs. retention probability) on the quality of the resulting maps, we wrote a `simulator` that mimics the actual experiment performed in the laboratory, using the accepted statistical model of the RH experiment (Lange et al. 1995; Ben-Dor and Chor 1997). The following assumptions and notations are used:

1. The number of copies (ploidity) of the chromosome in the cell line used to construct the RH panel is denoted by c. The cell line is either haploid ($c = 1$) or diploid ($c = 2$).
2. The number of markers is denoted by n. We assume that the markers are uniformly and independently distributed along the chromosome.
3. The number of hybrids is denoted by m. It is assumed that events on different hybrids are mutually independent.
4. We assume that the breakpoints along each copy of the chromosome, induced by the radiation, are distributed according to a Poisson process, with rate λ.
5. We assume that the retention probability of every chromosome fragment is the same and denote its value by p. We use q to denote $1 - p$.
6. Each of the mn hybrid/marker typings is subject to false-positive and/or false-negative errors with probability α and β, respectively.
7. The typing process is done either once or twice for each hybrid/marker pair. We denote the number of typing by ty ($ty = 1$ or $ty = 2$). If $ty = 2$ and the two typings for the same hybrid/marker pair differ, then the result of this pair is denoted by "?" and is considered ambiguous. (Two typings are used to reduce false-positive and/or false-negative errors.)
8. All processes (markers distribution, breakage, retention, and typing errors) are independent.

The input to the `simulator` are the experiment parameters

$c,n,m,\lambda,p,\alpha,\beta,t$. The output of the `simulator` (like a real RH laboratory experiment) is an $n \times m$ matrix, D, which depicts the typing results for each marker and hybrid: "+" for a retained marker, "−" for a lost markers, and "?" for an ambiguous typing.

Reduction to TSP

The TSP is an NP-complete combinatorial optimization problem (Garey and Johnson 1979; Corman et al. 1990) that has been studied extensively. In this problem, a salesman has to visit n cities exactly once and then return to the first city. The goal is to find a tour that minimizes the total cost of the tour (sum of costs to travel along each edge of the tour). In this section we describe the `evaluator`, which reduces the problem of markers' ordering to the TSP [this approach was suggested by Karp et al. (1996)]. Because, in practice, the number of typed markers is too large to be reliably ordered, an optional pruning step is first performed. In this step, the `evaluator` chooses a representative set of framework markers, F, which is spaced "fairly evenly." We detail the pruning process below. After the pruning step, two TSP instances are produced, corresponding to the two optimization criteria. In both cases the output of the `evaluator` is a complete undirected weighted graph, G, whose nodes are all the markers and an additional `start` vertex, s. The weight of an edge from i to j in G depends on the retention pattern of the corresponding markers and on the optimization criteria.

Reducing OCB to TSP

The reduction is performed as follows [a similar approach (called gap-minimization) was used by Alizadeh et al. (1995) for STS-content mapping]. (1) The distance between the `start` vertex s and all other vertices in set to zero. (2) The distance between marker i and marker j is set to the ratio of separating hybrids to informative hybrids. A separating hybrid is one in which marker i was retained but marker j was lost, or vice versa. An informative hybrid is a hybrid that has nonambiguous typings for both markers.

If the RH data does not contain ambiguous entries, then each of the m hybrids is informative with respect to every pair of markers. In this case, the number of OBCs induced by any permutation equals exactly m times the weight of the corresponding tour in G (Ben-Dor and Chor 1997). Thus, finding the minimum length tour is equivalent to optimizing with respect to the OCB criteria.

However, if the matrix D does contains "?" entries, this reduction is not always valid. The reason is that in the presence of "?" entries, the OCB score of a permutation depends on the retention pattern of the nonadjacent markers on the two sides of the "?" entry. For example, consider the following five hybrid, three-marker matrix:

Marker	*A*	*B*	*C*
Hybrid 1	+	?	+
Hybrid 2	+	?	+
Hybrid 3	+	?	+
Hybrid 4	−	−	+
Hybrid 5	−	+	+

and let G denote the resulting graph. Consider now the permutations $\pi = A,B,C$ and $\tau = A,C,B$. On one hand, π induces fewer breakpoints than τ (two vs. three). On the other hand, the tour τ is shorter than the tour π (0.5 + 0.4 = 0.9 vs. 0.5 + 0.5 = 1). Therefore, the reduction is not valid in this case.

Even though the reduction is not valid in the general case, we claim that because in practice the fraction of "?" is typically low (~1%), the total length of a tour is generally close to the OCB count of the corresponding permutation. We can use this reduction to solve an approximation of the OCB problem.

Reducing MLE to TSP

The reduction is performed in three steps: First, the retention probability, p, is estimated (we assume that the negative and positive error rates are estimated in the lab and are given as part of the input). Then, the breakage probability between every pair of markers is estimated. Finally, weights to the graph edges are assigned such that the total length of a tour equals the minus logarithm of the likelihood of the corresponding permutation. Therefore, minimizing the total tour length is equivalent to maximizing the logarithm of the likelihood of the permutation. As the logarithm is a monotonically increasing function, this is the same as maximizing the likelihood of the permutation.

We remark that this reduction is valid only for haploid, error-free data (i.e., a Markovian model). In non-Markovian models, the retention pattern of a marker depends on the retention pattern of all previous markers. For non-Markovian models, the likelihood calculation is much harder. First, the most likely breakage probabilities depend on the markers' permutation. Thus, these probabilities should be estimated from scratch for every proposed permutation. Second, even for a fixed permutation, the breakage probabilities between adjacent markers are not independent and need to be estimated together using iterative methods (e.g., the EM algorithm).

We use the following "hybrid" approach. In the first step, we estimate the breakage probabilities between every pair of markers assuming the general model (i.e., taking into account diploidity and laboratory errors). Then, we treat the RH data as if it was produced by a Markovian model, reduce the MLE problem to TSP (as explained below), and solve the resulting TSP instance. The MLE objective function is altered by the reduction for non-Markovian models, but good TSP permutations proved to be good MLE permutations for both real and synthetic data. At the final stage, the likelihood of the resulting permutation is evaluated in the full model (using EM).

We now describe the reduction in details. The retention probability, p, is estimated by the ratio of the total number of "+" entries to the total number of nonambiguous entries. The estimation of the breakage probabilities $\{\theta_{i,j}\}$, for every pair of markers is more involved as it depends on the retention patterns of the two markers as well as on the model characteristics (i.e., error rates, ploidity, number of typings). It is described below.

Estimating the Breakage Probability

In this section we describe how to compute the most likely estimation of the breakage probabilities $\{\theta_{i,j}\}$ between two markers, i and j (a similar analysis is given by Slonim (1996). Consider first a single hybrid, and let O_i denote the observation of marker i in the hybrid, that is, $O_i \in \{-,+\}$ if only one typing is performed, and $O_i \in \{-,+,?\}$ if double typing is done. Let S_i denote the number of copies of marker i retained in the

hybrid ($S_i \in \{0, 1\}$ for a haploid panel, whereas $S_i \in \{0, 1, 2\}$ for a diploid panel).

The observation probability, $\text{Obs}(O_i \mid S_i)$, is the probability of observing O_i given that the state is S_i. This probability depends on the error rates (α and β), and on *ty* (whether the typing was performed once or twice). The observation probabilities for $ty = 1$ are given by

S_i \ O_i	−	+
0	$1 - \alpha$	α
1, 2	β	$1 - \beta$

Whereas for double typing these probabilities are given by

S_i \ O_i	−	+	?
0	$(1 - \alpha)^2$	α^2	$2\alpha(1 - \alpha)$
1, 2	β^2	$(1 - \beta)^2$	$2\beta(1 - \beta)$

The costate probability, State $(\langle S_i, S_j\rangle \mid \theta)$ is the probability of being in state S_i at marker *i* and in state S_j at marker *j*, given that the breakage probability is θ. This probability depends on the retention probability, *p*, and on the ploidity of the panel. For example, if the panel is haploid, the probability that both markers are retained in the hybrid ($S_i = S_j = 1$), satisfy

$$\text{State}(\langle S_i, S_j\rangle = \langle 1, 1\rangle \mid \theta) = \Pr(S_i = 1) \cdot \Pr(S_j = 1 \mid S_i = 1, \theta)$$

Marker *i* is retained with probability *p*. Given that marker *i* is retained, marker *j* will also be retained unless a breakpoint occur (with probability θ), and the fragment containing marker *j* will be lost (with probability $q = 1 - p$). Thus, $\text{State}(\langle S_i, S_j\rangle = \langle 1, 1\rangle \mid \theta) = p(1 - \theta q)$. The costate probabilities for the haploid case are given by

S_i \ S_j	0	1
0	$q(1 - \theta p)$	θpq
1	θpq	$p(1 - \theta q)$

Let us denote the above probabilities by S_{00}, S_{01}, S_{10}, and S_{11}. The diploid case is derived easily by considering the independent fates of the individual fragments. The costate probabilities, $\text{State}(\langle S_i, S_j\rangle \mid \theta)$ in the diploid case are given by

S_i \ S_j	0	1	2
0	$(S_{00})^2$	$2S_{00}S_{01}$	$(S_{01})^2$
1	$2S_{10}S_{01}$	$S_{00}S_{11} + S_{10}S_{01}$	$(S_{01})^2$
2	$(S_{10})^2$	$2S_{10}S_{11}$	$(S_{11})^2$

The probability of observing $\langle O_i, O_j\rangle$ as a function of θ can be expressed as follows:

$$\begin{aligned}\Pr(\langle O_i, O_j\rangle \mid \theta) &= \sum_{S_i,S_j} \Pr(\langle O_i, O_j\rangle \mid \langle S_i, S_j\rangle, \theta) \cdot \Pr(\langle S_i, S_j\rangle \mid \theta) \\ &= \sum_{S_i,S_j} \text{State}(\langle S_i, S_j\rangle \mid \theta) \cdot \text{Obs}(O_i \mid S_i) \\ &\quad \cdot \text{Obs}(O_j \mid S_j)\end{aligned}$$

Because the different hybrids are independent of each other, the probability of observing the data is the product of the probability of observing each hybrid. Let n_{ij}^{++} denote the number of hybrids that contain both markers. Let n_{ij}^{+-} denote the number of hybrids that contain markers *i* but not marker *j*. Similarly define n_{ij}^{-+}, and n_{ij}^{--}. (Hybrids with ambiguous typing for either marker *i* or marker *j* are not counted. In cases where α and β are on the same order of magnitude, these hybrids supply little of no additional information.) Using these notations and the probability for one hybrid (equation 1, below), we obtain

$$\begin{aligned}\Pr(n_{ij}^{--}, n_{ij}^{-+}, n_{ij}^{+-}, n_{ij}^{++} \mid \theta) = \Pr(\langle -, -\rangle \mid \theta)^{n_{ij}^{--}} \\ \cdot \Pr(\langle -, +\rangle \mid \theta)^{n_{ij}^{-+}} \\ \cdot \Pr(\langle +, -\rangle \mid \theta)^{n_{ij}^{+-}} \\ \cdot \Pr(\langle +, +\rangle \mid \theta)^{n_{ij}^{++}} \qquad (1)\end{aligned}$$

The value assigned to θ_{ij} is the value of θ that maximizes the above expression. Solving this maximization problem amounts to finding the roots of its derivative, which is a polynomial in θ. The degree of the polynomial depends on the ploidity of the panel. For a haploid panel, the polynomial has degree 2, and thus the roots are calculated analytically. For the diploid case, however, the polynomial has degree 5, and the roots are computed using numerical methods. We omitted this polynomial from the text, as it takes more than a page to display and offers no new insight. We remark that for the WI RH (diploid) data, using the high degree polynomial did not improve the quality of the resulting permutation compared with using the quadratic polynomial.

We now describe in detail the computation for the haploid, error-free model (i.e., $\alpha = \beta = 0$). In this case, equation 1 reduces to

$$\begin{aligned}\Pr(n_{ij}^{--}, n_{ij}^{-+}, n_{ij}^{+-}, n_{ij}^{++} \mid \theta) = (q(1 - \theta p))^{n_{ij}^{--}} (\theta pq)^{(n_{ij}^{-+} + n_{ij}^{+-})} \\ (q(1 - \theta q))^{n_{ij}^{++}} \qquad (2)\end{aligned}$$

By equating the derivative to 0, we get that $\theta_{i,j}$ is

$$\theta_{i,j} = \frac{B - \sqrt{B^2 - 4AC}}{2A} \qquad (3)$$

where

$$A = pqm,\ B = pn_{ij}^{--} + qn_{ij}^{++} + n_{ij}^{-+} + n_{ij}^{+-}, \text{ and } C = n_{ij}^{+-} + n_{ij}^{-+}$$

Because $\theta_{i,j}$ represents a probability, we trim it to the range [0, 1] if needed.

Assigning Weights to the Graph Edges

After the optimal breakage probabilities are estimated, we can reduce the likelihood maximization problem to TSP (Karp et al. 1996). We assign a transition probability, $\text{Trans}(e)$, to every edge in *G*.

1. The transition probability of the edge (s,i) between the `start` vertex and the *i*th marker is set to

$$\text{Trans}(s,i) = \left(\sqrt{p}\right)^{n_i^+} \left(\sqrt{q}\right)^{n_i^-}$$

where n_i^+ is the number of hybrids that retain marker *i*; n_i^- is the number of hybrids that did not retain this marker.

2. The transition probability of the edge (i,j) between marker i and marker j is set to

$$\text{Trans}(i,j) = (1-\theta_{i,j}p)^{n^{--}_{i,j}}\ (1-\theta_{i,j}q)^{n^{++}_{i,j}}\ \left(\theta_{i,j}\sqrt{pq}\right)^{(n^{+-}_{i,j}+n^{-+}_{i,j})}$$

Let π be a markers' permutation, and let $\pi' = \langle s, \pi(1), \pi(2), \ldots, \pi(n), s\rangle$ denote the tour in G corresponding to π. The following property is shown by Karp et al. (1996) for the haploid, error-free model: For every permutation π, the product of the transition probability of the edges in π' equals the likelihood of π. That is, $L(\pi) = \Pi_{e\in\pi'}\text{Trans}(e)$. Therefore, if we set the weight of an edge e in G to $-\log$ Trans (e), then we have

$$\sum_{e\in\pi'} weight(e) = -\log \Pr(D|\pi)$$

Thus finding a tour of minimum weight (TSP) is the same as finding a permutation with maximum likelihood (MLE). After an order is produced, its likelihood can be recalculated in the full model by using the EM algorithm. Specifically, we have used the EM engine implemented in RHMAPPER.

Pruning

In practice, the ratio of markers to hybrids is much larger than could be correctly ordered. Hence, an optional pruning step is performed. In this step, a sufficiently spaced set of framework markers, F, is chosen. The pruning process is controlled by a threshold parameter, θ_0, that allows us to control the minimal distance between framework markers. It is performed as follows: (1) Initially, F contain all the n markers. (2) We order the n markers at random. Then we go over the list, choosing one marker at a time. If the marker currently chosen is still in F (meaning it has not been removed), then we remove from F all markers that are "too close" to it. We estimate the breakage probability between the current marker and all other markers in F and remove all those whose estimated breakage probability is below θ_0.

Simulated Annealing

The *Simulated Annealing* algorithm (Press et al. 1992) is described in Figure 7. In essence, this is a local-search heuristic that combines the "hill-climbing" heuristic with a "random-walk" mechanism to help avoid local-minima traps. As any local-search algorithm, a current solution is kept in each iteration of the run. In this case, the solution is a permutation, P, together with its associated cost. The goal is to minimize the total cost of the tour.

At each iteration a new solution, P', is chosen according to the neighborhood structure (described below). At the start of the run, "bad" neighbors P' are allowed. That is, neighbors that substantially increase the objective function (the cost of the permutation) may be chosen. As time passes, the probability that such steps are taken decreases, as dictated by the system "temperature." The exact value of the temperature and the halting conditions are governed by a cooling scheme. Several cooling schemes were tested, and the best one was chosen in our implementation. It is described below. At the end of the algorithm, the best permutation is returned.

Neighborhood Structure

This concept is essential in any local search algorithm, and simulated annealing is no exception. Given a state in the search space, it defines the set of "eligible" states that the algorithm may proceed to. Here, the state is a permutation P of the numbers $\{1...n\}$. We implemented the *2-OPT* neighborhood structure (Lin and Kernighan 1973). In this structure, two permutations π and τ are neighbors if and only if τ can be obtained from π by subdividing π into three parts and reversing the middle part, that is, if and only if π can be written as ABC and τ as AB^RC, where B^R is the reversal of B.

The efficiency of computing the expected difference between the cost of the current permutation and the cost of a proposed permutation has large impact on the efficiency of the search and thus on the quality of the final solution. One of the main advantages of reducing general optimization problems (such as OCB and MLE) to TSP lies in the ease of computing this cost difference for TSP. In the *2-OPT* neighborhood structure, this computation can be done in constant time, replacing two edges by two others.

Cooling Scheme

We have tested the algorithm extensively with different choices of cooling schemes on synthetic RH data and implemented the best one in RHO. It is the RHmap cooling scheme from RHMAP (Boehnke et al. 1996), which is characterized by five parameters: *T_0, decay_factor* $\in[0..1]$, *nmove, nbet,* and *ntemp*. The initial temperature is *T_0*. When either *nbet* moves improve the objective function or *nmove* moves of any type are performed, the current temperature is decreased—it is multiplied by *decay_factor*. The algorithm stops when a total of *ntemp* temperature lowerings were done. The parameters used are as following: *T_0* = 0.5, *nmove* = $15n$, *nbet* = $5n$, *decay_factor* = 0.9, and *ntemp* = n (where n denotes the number of markers).

Held-Karp Method

For the sake of completeness, we briefly describe in this section the *Held-Karp* method (Held and Karp 1970, 1971) that we use to obtain tight lower bounds for the TSP. Let $C = \{c(i,j)\}$ be an $n \times n$ symmetric matrix that specifies the weights assigned to the edges of a complete undirected graph with vertex set $\{1, 2, \ldots, n\}$. A tour is a cycle passing through each vertex exactly once. We seek a tour of minimum weight. Let C^* denote the weight of a minimum tour. The *Held-Karp* method aims at finding tight lower bounds for C^*.

A tree is a connected graph without cycles. A 1 − tree is

1. $P \leftarrow$ a random permutation, or a user defined starting permutation.
2. Set T to the initial temperature.
3. Loop until any halt condition is met:
   - Pick a random P-neighbour, P', according to the neighbourhood structure. Let Δ be the incurred change in the cost of the permutation.
   - If Δ is negative then Change P to P'. Otherwise (positive Δ), change P to P' with probability $e^{-\frac{\Delta}{T}}$.
   - Update the current temperature according to the cooling scheme.
   - Save P if is better the the current best solution.
4. Output the best permutation found.

Figure 7 Outline of *sa-tsp* algorithm.

a tree over the vertex set {2, 3, . . . , n}, together with two distinct edges that touch vertex 1 (thus, the total number of edges in a 1 − tree is n). A minimum-weight 1 − tree is easy to compute using a small modification of a standard MST algorithm (e.g., Kruskal 1956). We first find a MST for the graph induced by the vertices {2, . . . ,n}. Then, we add the two edges with minimal weights that touch vertex 1. The *Held–Karp* method exploits the following relationship between tours and 1 − trees: (1) A tour is simply a 1 − tree in which each vertex has degree exactly 2. (2) If a minimum-weight 1 − tree is a tour, then it is a tour of minimum weight. Let $X = x_1, \ldots, x_n$ be a real n-vector. Consider the edge weights, C_X, resulting by adding x_i to every edge that touches vertex i. That is,

$$C_X = \{c_x(i,j)\}, \text{ where } c_x(i,j) = c(i,j) + x_i + x_j$$

For every tour (or 1 − tree), T, let us denote by $C(T)$ the weight of T with respect to the original weights (i.e., C), and by $C_X(T)$ the weight of T with respect to the weights C_X. Note that any tour, T, satisfies

$$C_X(T) = C(T) + 2\sum_{i=1}^{n} x_i$$

because a tour goes through each vertex exactly twice.

Let T_X be a minimum weight 1 − tree with respect to the weights C_X. Note that $C_X(T_X)$ is a lower bound for the weight of any tour with respect to the weights C_X. In particular, let Opt be an optimal tour with respect to the original weights (C(Opt) = C*). We have

$$C_X(T_X) \le C_X(\text{Opt} = C(\text{Opt})) + 2\sum_{i=1}^{n} x_i = C^* + 2\sum_{i=1}^{n} x_i$$

We denote by $w(X)$ the lower bound implied on C* by X, that is

$$w(X) = C_X(T_X) - 2\sum_{i=1}^{n} x_i \le C^*$$

Thus, we obtain an infinite family of lower bounds on the weight of an optimum tour, and the best such bound is $\sup_X w(X)$.

Approximating $\sup_X w(X)$

In this section we present the iterative method of Held and Karp (1971) for approximating $\sup_X w(X)$. The *Held–Karp* algorithm is given a positive parameter, t, and is described in Figure 8.

The intuition behind the iteration is the following. Because T_X contains n edges, the average degree of a vertex in T_X is 2. If T_X happens to be a tour (and thus is an optimal tour), then all the degrees are exactly 2, and X has reached a fixed point. Otherwise, $d_i - 2$ represents the "distance" between the degree of vertex i in the T_X and its "desired" degree in a tour. If d_i>2, then x_i increases, and thus the weights of edges that touch vertex i are increased. As a result, in the next minimal-weight 1 − tree, the degree of mode i tends to decrease. On the other hand, if d_i<2 (i.e., node i is a leaf in T_X), then x_i decreases, and thus the weights of the edges that touch vertex i are decreased. As a consequence, the degree of this node tends to increase in the next 1 − tree. Thus, each iteration updates X according to the current topology of T_X, towards a tour (which is a fixed point for the iteration).

Held and Karp have proved various properties of this algorithm (Held and Karp 1971; theorem 1). They bound the size of the gap between the convergence point of the lower bounds produced by the iterations and the optimal bound. This gap depends linearly on t. Smaller values of t tend to produce tighter lower bounds. Thus, it is desirable to choose small values of t. On the other hand, t controls the rate of convergence. Smaller values of t tend to require a larger number of iterations to converge.

As noted by Held and Karp (1971), a possible improvement to the basic iteration would be to change the value of t over time. We implemented the following variant: Initially choose a large value t_1 and start iterating. After sufficiently many iterations, the process gets near the optimal bound minus the gap for t_1. Further iterating with the same value t_1 typically does not increase the lower bound.

At this stage we set t to t_2 (which is smaller than t_1) and continue the iterations with the new value of t, starting from the best X found so far. The new value of t enables the iteration to get closer to the optimal bound. We repeat with this process with decreasing values $t_3, t_4, \ldots, t_l$ and get values w_{ti} closer and closer to $\sup_X w(X)$.

When the iteration is applied to the TSP resulting from RH data, the 1 − tree produced tends to resemble tours very closely. Most vertices have degree 2. We use this 1 − tree to produce partial ordering of the markers. After removing the vertex 1 (which corresponds to the *start* vertex, which was introduced in the reduction to TSP) and the two edges that touch it from the 1 − tree, we are left with a tree, Tr, over the set of markers. Because the 1 − tree resembles a tour, Tr resembles a path. Now we proceed as suggested by the *MST-Order* algorithm (Ben-Dor and Chor 1997)—we find the longest path in the tree Tr and use this path to order the markers along it.

After implementing the basic algorithms, we tested it extensively on synthetic data. In all cases the quality of the permutation returned by the *Simulated Annealing* algorithm was very close to the lower bound implied by the *Held–Karp* algorithm. Moreover, the partial ordering returned by the *Held–Karp* algorithm contained most of the markers and, in many cases, all of them. In these cases we have provably found the optimal solution to the corresponding TSP.

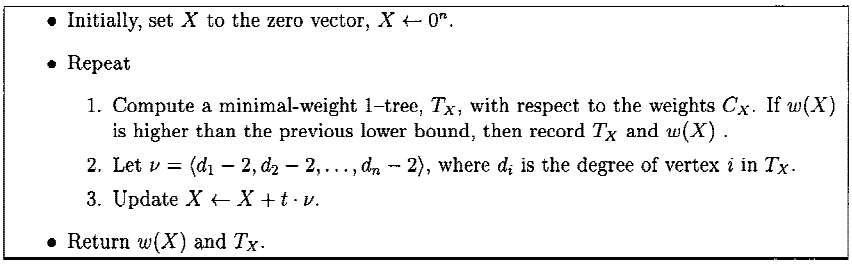


Figure 8 Outline of *Held–Karp* algorithm.

ACKNOWLEDGMENTS

We thank Ann Becker, Micha Skala, Arie Tal, and Roman Talyansky who implemented parts of the software, and David Cox, Dan Gusfield, Richard Karp, Ron Shamir and Zohar Ya-

chini for very helpful and stimulating discussions. We thank the anonimous referees. This work was supported by the Fund for Promotion of Research at the Technion (B.C.).

The publication costs of this article were defrayed in part by payment of page charges. This article must therefore be hereby marked "advertisement" in accordance with 18 USC section 1734 solely to indicate this fact.

REFERENCES

Alizadeh, F., R. Karp, D. Weisser, and G. Zweig. 1995. Physical mapping of chromosomes using unique probes. *J. comp. Biol.* **2:** 159-184.

Barrett, J.H. 1992. Genetic mapping based on radition hybrid data. *Genomics* **13:** 95–103.

Ben-Dor, A. and B. Chor. 1997. On constructing radiation hybrid maps. *J. Comp. Biol.* **4:** 517-533.

Ben-Dor, A. and D. Pelleg. 1998. Laboratory for Computational Biology Research Page. http://www.cs.technion.ac.il/Labs/cbl/research.html. Computer Science department, Technion, Haifa, Israel

Bishop, D.T. and G.P. Crockford. 1992. Comparisons of radiation hybrid mapping and linkage mapping. *Cytogenet. Cell Genet.* **59:** 93-95.

Boehnke, M. 1992. Radiation hybrid mapping by minimization of the number of obligate chromosome breaks. *Cytogenet. Cell Genet.* **59:** 96–98.

Boehnke, M., K. Lange, and D. Cox. 1991. Statistical methods for multipoint radiation hybrid mapping. *Am. J. Hum. Genet.* **49:** 1174–1188.

Boehnke, M., K. Lunetta, E. Hauser, K. Lange, J. Uro, and J. VanderStoep. 1996. RHMAP: Statistical Package for Multipoint Radiation Hybrid Mapping, version 3.0. Software and documentation available from http://www.spn.umich.edu/group/statgen/software.

Chor, B. and M. Sudan. 1998. A geometric approach to betweenness. *SIAM J. Discrete Math.* **11:** 511–523.

Corman, T., C. Leiserson, and R. Rivest. 1990. *Introduction to algorithms*. MIT Press, Cambridge, MASS.

Cox, D., M. Burmeister, E. Price, S. Kim, and M. Myers. 1990. Radiation hybrid mapping: A somatic cell genetic method for constructing high-resolution maps of mammalian chromosomes. *Science* **250:** 245–250.

Garey, M. and D. Johnson. 1979. *Computers and intractability, a guide to the to theory of NP-completeness*. W.H. Freeman, New York, NY.

Goss, S. and H. Harris. 1975. New method for mapping genes in human chromosomes. *Nature* **255:** 680–684.

———. 1977a. Gene transfer by means of cell fusion. I. Statistical mapping of the human X-chromosome by analysis of radiation-induced gene segregation. *J. Cell Sci.* **25:** 17–37.

———. 1977b. Gene transfer by means of cell fusion. II. The mapping of 8 loci on human chromosome 1 by statistical analysis of gene assortment in somatic cell hybrids. *J. Cell Sci.* **25:** 39–57.

Held, M. and R. Karp. 1970. The traveling-salesman problem and minimum spanning trees: Part I. *Operation Res.* **18:** 1138–1162.

———. 1971. The traveling-salesman problem and minimum spanning trees: Part II. *J. Math. Prog.* **1:** 6–25.

Hudson, T., L. Stein, S. Gerety, J. Ma, A. Castle, J. Silva, D. Slonim, R. Baptista, L. Kruglyak, S. Xu et al. 1995. An STS-based map of the human genome. *Science* **270:** 1945–1954.

Hui, L. 1997. Radiation Hybrid Map Data Files. http://www.genome.wi.mit.edu/ftp/distribution/human_STS_releases/july97/rhmap/README.html. Whitehead Institute/MIT Center for Genome Research.

Johnson, C., R. Singer, and J. Lawrence. 1991. Fluorescent detection of nuclear RNA and DNA: Implications for genome organizations. *Meth. Cell Biol.* **35:** 73–98.

Karp, R., W. Ruzzo, and M. Tompa. 1996. Algorithms in molecular biology - lecture notes. Department of Computer Science and Engineering, University of Washington, Seattle, WA.

Kruskal, J. 1956. On the shortest spanning subtree of a graph and the traveling salesman problem. *Proc. Am. Math. Soc.* **7:** 48–50.

Lange, K. and M. Boehnke. 1992. Bayesian methods and optimal experimental design for gene mapping by radiation hybrids. *Ann. Hum. Genet.* **56:** 119–144.

Lange, K., M. Boehnke, D. Cox, and K. Lunetta. 1995. Statistical methods for polyploid radiation hybrid mapping. *Genome Res.* **5:** 136–150.

Lin, S. and B.W. Kernighan. 1973. An efficient heuristic algorithm for the traveling-salesman problem. *Operation Res.* **21:**(2)

Lunetta, K. and M. Boehnke. 1994. Multipoint radiation hybrid mapping: Comparison of methods, sample size requirements, and optimal study characteristics. *Genomics* **21:** 92–103.

Lunetta, K., M. Boehnke, K. Lange, and D. Cox. 1995. Experimental design and error detection for polyploid radiation hybrid mapping. *Genome Res.* **5:** 151–163.

Manly, K.F. and J.M. Olson. 1999. Overview of QTL mapping software and introduction to Map Manager QTL. *Mamm. Genome* 327–334.

Matise, T., M. Perlin, and A. Chakravarti. 1994. Automated construction of genetic linkage maps using an expert system (MultiMap): A human genome linkage map. *Nat. Genetics* **6:**(4) 384–390.

Newell, W., R. Mott, S. Beck, and H. Lehrach. 1995. Construction of genetic maps using distance geometry. *Genomics* **30:** 59–70.

Newell, W., S. Beck, H. Lehrach, and A. Lyall. 1998. Estimation of distances and map construction using radiation hybrids. *Genome Res.* **8:** 493–508.

Opatrny, J. 1979. Total ordering problems. *Siam J. Comput.* **8:**(1) 111–114.

Press, W., B. Flannery, S. Teukolsky, and W. Vetterling. 1992. *Numerical recipes. The art of scientific computing*, 2nd edition. Cambridge University Press, Cambridge, UK.

Rabiner, L. 1989. A tutorial on hidden markov models and selected applications in speech recognition. *Proc. Inst. Electr. Electron. Eng.* **77:** 257–286.

Slonim, D. 1996. "Learning from imperfect data in theory and practice." Ph.D. thesis, Massachusetts Institute of Technology, Cambridge, MA.

Slonim, D., L. Stein, L. Kruglyak, and E. Lander. 1996. RHMAPPER: An interactive computer package for constructing radiation hybrids maps. http://www.genome.wi.mit.edu/ftp/pub/software/rhmapper/.

Stein, L., L. Kruglyak, D. Slonim, and E. Lander. 1997. Building human genome maps with radiation hybrids. In *RECOMB*, pp. 277–286.

Stewart, E., K. McKusick, A. Aggarwal, E. Bajorek, S. Brady, A. Chu, N. Fang, D. Hadley, M. Harris, S. Hussain et al. 1997. An STS-based radiation hybrid map of the human genome. *Genome Res.* **7:** 422–433.

Thompson, E. 1987. Crossover counts and likelihood in multipoint linkage analysis. *IMA J. Math. Appl. Med. Bio.* **4:** 93–108.

Walter, M., D. Spillett, P. Thomas, W. J., and P. Goodfellow. 1994. A method for constructing radiation hybrid maps of whole genomes. *Nat. Genet.* **7:** 22–28.

Weeks, D., T. Lehner, and J. Ott. 1992. Preliminary ranking procedures for multilocus ordering based on radiation hybrid data. *Cytogenet. Cell Genet.* **59:** 125–127.

Received August 9, 1999; accepted in revised form January 20, 2000.