Chromosome mapping: Radiation hybrid data and stochastic spin models

C. T. Falk[1,*] and H. Falk[2,†]
[1]*The New York Blood Center, New York, New York 10021*
[2]*Physics Department, City College of the City University of New York, New York, New York 10021*
(Received 4 November 1994)

This work approaches human chromosome mapping by developing algorithms for ordering markers associated with radiation hybrid data. Motivated by the recent work of Boehnke, Lange, and Cox [Am. J. Hum. Genet. **49**, 1174 (1991)], we formulate the ordering problem by developing stochastic spin models to search for minimum-break marker configurations. In one particular application, the methods developed are applied to 14 human chromosome-21 markers tested by Cox *et al.* [Science **250**, 245 (1990)]. The methods generate configurations consistent with the best ones found by others. Additionally, we find that the set of low-lying configurations is described by a Markov-like ordering probability distribution. The distribution displays cluster correlations reflecting closely linked loci.

PACS number(s): 87.10.+e, 02.70.−c, 05.20.−y

I. INTRODUCTION

The use of data from radiation hybrid (RH) experiments has become a useful method for fine structure mapping of human chromosomes. Based on methods described by Goss and Harris [1,2], Cox *et al.* [3], and Burmeister *et al.* [4] have developed the technique in detail so that results from their experiments provide material for ordering DNA markers on human chromosomes.

The basic strategy employed in radiation hybrid mapping (fully described in Cox *et al.* [3]) entails irradiating a rodent-human somatic cell hybrid, which contains a particular human chromosome, with a lethal dose of x rays. This will cause the chromosomes to break into several fragments. After fusion with HPRT-deficient rodent cell lines, only the fused cells, containing both the x-ray irradiated cells and the normal rodent cells, will survive if grown in HAT medium. Detailed descriptions of HPRT and HAT are contained in Ref. [3]. Each hybrid clone arising from this fusion will contain a unique set of fragments from the original human chromosome. Each clone can then be typed for a set of human DNA markers (equivalently, loci) known to be on that human chromosome. Based on the assumption that tightly linked markers are unlikely to be broken apart by the radiation, markers close to one another will show a correlated pattern of retention in the clones; whereas, distant markers will be retained in a relatively independent manner.

Several methods for ordering markers have been developed using results from RH experiments [including both parametric (Cox *et al.* [3], Boehnke *et al.* [5]) and nonparametric methods (Boehnke *et al.* [5], Falk [6], Bishop and Crockford [7], Weeks *et al.* [8]). In particular, Boehnke *et al.* [5] used a mathematical quantity associated with the number of breaks and then used optimization techniques to minimize that quantity. One optimization technique involved a simulated annealing search for configurations associated with minimal numbers of breaks.

We set out to study and understand the work of Boehnke *et al.*, and we developed a formulation in the context of stochastic spin systems (Falk [9]). It may be useful to point out some differences in the formulations.

Boehnke *et al.* use block inversions of a given marker order and compare the old and new orders with respect to "obligate" breaks. They then apply simulated annealing techniques and decide, at each time step, whether to retain the original order or transition to the new. If a transition would result in a smaller number of breaks, the transition is made with probability 1. If a transition would not decrease the number of breaks, then the transition probability is less than 1, and that transition probability systematically decreases over time.

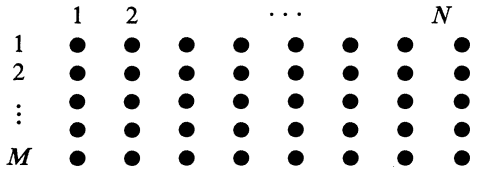
In our study we implement three algorithms which incorporate the number of "breaks." The three algorithms are three stochastic spin models. These, too, are designed to search for configurations with small numbers of breaks. A probability is constructed to determine whether or not to retain the current order or transition to the new. The probabilities are set up so as to bias the decision towards transitions to configurations with fewer breaks; however, at a given step, a possible transition leading to a smaller number of breaks will not necessarily be realized.

The spin language provides mathematically intuitive expressions for the number of breaks. Those expressions contain products of adjacent spin variables, and calculations involving breaks are easily presented in spin notation. For those seeking a rigorous mathematical setting, we remark that Liggett [10] has treated stochastic spin models and related models from biology, physics, and economics. Liggett's book contains an extensive bibliography and guides the reader to survey papers by Griffeath, Durrett, Stroock, Holley, and others.

*Electronic address: cathy@server.nybc.org
†Electronic address: alfcc@cunyvm.cuny.edu

II. METHODS

We are considering M cones, each of which is tested for the presence (or absence) of N different DNA markers. It is convenient to represent the clones as M rows, each with N sites. Thus, one pictures a two-dimensional array of M rows, N columns:



Assign a particular DNA marker, labeled f_j, $(j=1,2,\ldots,N)$, to each column. Associate a variable ("a spin") s_{ij} with the site at row i, column j. If the marker f_j is present at site (i,j), take $s_{ij}=+1$; if f_j is not present at site (i,j), take $s_{ij}=-1$. If one is uncertain as to whether a marker is present at site (i,j), we take s_{ij} to be unknown, and we deal with such sites in a manner to be specified subsequently.

For $(j=1,2,\ldots,N)$ the marker f_j was arbitrarily assigned to column j. But any of the $N!$ assignments would be possible. A criterion is needed for judging the "goodness" of an assignment.

Considering the DNA markers being tested for lie on a particular human chromosome, those markers which are tightly linked are likely to appear together or not appear in each clone. Therefore, it is reasonable to seek those marker assignments which reflect such correlation. For that purpose we say that a "break" exists between sites (i,j) and $(i,j+1)$ whenever $s_{ij}s_{i,j+1}=-1$. The strategy is to minimize the total number of breaks. Note that in the two-dimensional array of spins, a break refers only to horizontal, nearest-neighbor spin pairs.

Here we devise and test several algorithms which monitor the total number of breaks while selectively permuting column labels. The algorithms attempt to explore the vast configuration space in the manner of a "random walk," biased toward configurations having a reduced number of breaks. Notice that for $N=14$ there are $N!=87\,178\,291\,200$ permutations of the numbers $(1,2,\ldots,14)$. Thus, in the spirit of the traveling salesman problem, simulated annealing techniques are used (Kirkpatrick *et al.* [11], Press *et al.* [12]).

A. Nearest-neighbor transposition algorithm

(0) Start with a specified configuration $\{s\}$ of the variables s_{ij}.

(1) Select a number j at random from the set $\{1,2,\ldots,N\}$.

(2) If $j\neq N$, compute the total number of breaks between columns $j-1,j$ and between columns $j+1,j+2$. Denote that number by $B_j(1)$:

$$B_j(1)=\sum_{q=1}^{M}\left[(1-\delta_{j,1})\frac{1-s_{q,j-1}s_{qj}}{2}+(1-\delta_{j,N-1})\frac{1-s_{q,j+1}s_{q,j+2}}{2}\right] \tag{2.1}$$

for $j\in\{1,2,\ldots,N-1\}$. The Kronecker delta contained in $(1-\delta_{i,j})$ is inserted to handle "end effects" since column 1 has no left neighbor and N has no right neighbor.

(3) Repeat (2) with the spin values s_{ij} and $s_{i,j+1}$ interchanged for $i=1,2,\ldots,M$, and denote the resulting sum by $B_j(2)$ instead of $B_j(1)$.

(4) Then compute B_j, the change in the number of breaks resulting from the interchange of the columns of spin values s_{ij} and $s_{i,j+1}$.

$$B_j=B_j(2)-B_j(1) \quad \text{for } j\in\{1,2,\ldots,N-1\}\ . \tag{2.2}$$

(5) Interchange DNA marker column assignments and the associated spin values for columns j and $j+1$ with probability

$$w_{1j}=\begin{cases}\dfrac{\exp(-\beta B_j/M)}{2\cosh(\beta B_j/M)} & \text{for } j\in\{1,2,\ldots,N-1\}\\ 0 \quad \text{for } j=N\ .\end{cases} \tag{2.3}$$

Do not interchange DNA marker column assignments and the associated spin values for columns j and $j+1$ with probability

$$w_{2j}=1-w_{1j}\ . \tag{2.4}$$

Note

$$w_{2j}=\begin{cases}\dfrac{\exp(+\beta B_j/M)}{2\cosh(\beta B_j/M)} & \text{for } j\in\{1,2,\ldots,N-1\}\\ 1 \quad \text{for } j=N\ .\end{cases} \tag{2.5}$$

The "inverse temperature" parameter β $(0\leq\beta<\infty)$ is allowed to increase in an empirically determined manner as the algorithm is implemented. Clearly very large β strongly favors transitions reducing the number of breaks, whereas, a small, positive value of β makes transitions to increase or to decrease the number of breaks almost equally likely. Why not just take β large at the outset? The answer is (Kirkpatrick *et al.* [11], Press *et al.* [12]) that setting β large early in the calculation may cause the algorithm to get "trapped" in a local minimum before performing a significant search of configuration space.

Thus, the initially chosen value for β and the manner of increasing that value constitute the "art" of simulated annealing.

(6) Return to (1) and repeat the procedure starting with the current configuration $\{s'\}$ of the variables s'_{ij}. The procedure may be terminated after a specified number of iterations and/or when the total number of breaks

$$B=\sum_{q=1}^{M}\sum_{j=1}^{N-1}\frac{1-s_{qj}s_{q,j+1}}{2} \tag{2.6}$$

has realized acceptably small values. For each run one retains the configurations associated with the smallest values of B.

If β were fixed, then the above procedure would define a finite-state, discrete-time Markov chain with transition probability $p(\{s'\}|\{s\})$ from a configuration $\{s\}$ to

configuration $\{s'\}$. Explicitly

$$p(\{s'\}|\{s\})=\frac{1}{N}\sum_{j=1}^{N}\left\{\left[\prod_{q=1}^{M}\left[\frac{1+s'_{qj}s_{q,j+1}}{2}\right]\left[\frac{1+s'_{q,j+1}s_{qj}}{2}\right]\prod_{\substack{k=1\\k\neq j,j+1}}^{N}\left[\frac{1+s'_{qk}s_{qk}}{2}\right]\right]w_{1j}+\left[\prod_{q=1}^{M}\prod_{k=1}^{N}\left[\frac{1+s'_{qk}s_{qk}}{2}\right]\right]w_{2j}\right\}. \tag{2.7}$$

B. Two-column transposition algorithm

A natural extension of the nearest-neighbor transposition algorithm involves columns k,j with $k>j+1$. Remove two numbers at random from the set $\{1,2,\ldots,N\}$. Denote the smaller number by j and the larger by k. If $k=j+1$, follow the previously described nearest-neighbor algorithm, starting with step (2). If $k>j+1$, proceed as follows.

Consider the total number of breaks between columns $j-1,j;j,j+1;\ k-1,k;k,k+1$. Denote that number by $B_{jk}(1)$, where

$$B_{jk}(1)=\sum_{q=1}^{M}\left[(1-\delta_{j,1})\frac{1-s_{q,j-1}s_{qj}}{2}+\frac{1-s_{qj}s_{q,j+1}}{2}+\frac{1-s_{q,k-1}s_{qk}}{2}+(1-\delta_{k,N})\frac{1-s_{qk}s_{q,k+1}}{2}\right]. \tag{2.8}$$

After interchanging spin values s_{ij} and s_{ik} for $i=1,2,\ldots,M$, the number of breaks is

$$B_{jk}(2)=\sum_{q=1}^{M}\left[(1-\delta_{j,1})\frac{1-s_{q,j-1}s_{qk}}{2}+\frac{1-s_{qk}s_{q,j+1}}{2}+\frac{1-s_{q,k-1}s_{qj}}{2}+(1-\delta_{k,N})\frac{1-s_{qj}s_{q,k+1}}{2}\right]. \tag{2.9}$$

Then the change (in the number of breaks) resulting from the interchange is denoted by B_{jk}, where

$$B_{jk}=B_{jk}(2)-B_{jk}(1)\ . \tag{2.10}$$

Now interchange DNA marker column assignments and the associated spin values for columns j,k for $k>j+1$ with probability

$$w_{1jk}=\frac{\exp(-\beta B_{jk}/M)}{2\cosh(\beta B_{jk}/M)}\quad \text{for } j\in\{1,2,\ldots,N-2\}\ . \tag{2.11}$$

Do not interchange DNA marker column assignments and the associated spin values for columns j and k with probability

$$w_{2jk}=1-w_{1jk}\ . \tag{2.12}$$

C. Block-flip algorithm

As in the preceding algorithm, remove two numbers at random from the set $\{1,2,\ldots,N\}$. Denote the smaller number by j and the larger by k. If $k=j+1$, follow the previously described nearest-neighbor algorithm, starting with step (2). If $k>j+1$, proceed as follows.

The block consists of columns $j,\ldots,k$. Before flipping the block, the number of breaks involving columns $j-1,j$ and columns $k,k+1$ is

$$B_{jk}^{\text{block}}(1)=\sum_{q=1}^{M}\left[(1-\delta_{j,1})\frac{1-s_{q,j-1}s_{qj}}{2}+(1-\delta_{k,N})\frac{1-s_{qk}s_{q,k+1}}{2}\right]. \tag{2.13}$$

After flipping the block, the number of breaks is

$$B_{jk}^{\text{block}}(2)=\sum_{q=1}^{M}\left[(1-\delta_{j,1})\frac{1-s_{q,j-1}s_{qk}}{2}+(1-\delta_{k,N})\frac{1-s_{qj}s_{q,k+1}}{2}\right]. \tag{2.14}$$

Then the change in the number of breaks is denoted by B_{jk}^{block}, where

$$B_{jk}^{\text{block}}=B_{jk}^{\text{block}}(2)-B_{jk}^{\text{block}}(1)\ . \tag{2.15}$$

Now flip the block with probability

$$w_{1jk}^{\text{block}}=\frac{\exp(-\beta B_{jk}^{\text{block}}/M)}{2\cosh(\beta B_{jk}^{\text{block}}/M)}\quad \text{for } j\in\{1,2,\ldots,N-2\}\ . \tag{2.16}$$

Do not flip the block with probability

$$w_{2jk}^{\text{block}}=1-w_{1jk}^{\text{block}}\ . \tag{2.17}$$

The preceding method is similar to the block inversion algorithm used by Boehnke *et al.*, but our transition probabilities differ from theirs.

D. Unknown site content

In any clone one may have strings of one or more sites where the DNA marker associated with each site in a string is unknown. Thus, the spin values are unknown for the string. In the above algorithms such unknowns are dealt with in the following way.

Consider the case where the left and right ends of a string connect, respectively, to known spin s_{left} and to known spin s_{right}. If $s_{\text{left}}=s_{\text{right}}$, then all spins in the string are replaced by s_{right}. If $s_{\text{left}}\neq s_{\text{right}}$, then a fair coin toss is simulated. If the coin shows head (tail), then all spins in the string are replaced by $+1(-1)$.

If a string (of unknowns) contains an end spin, then re-

TABLE I. Fourteen chromosome-21 markers used in the examples. All numbered loci have a prefix of *D*21, APP denotes amyloid precursor, and SOD denotes superoxide dismutase.

APP	*S*1	*S*4	*S*8	*S*11	*S*12	*S*16	*S*18	*S*46	*S*47	*S*48	*S*52	*S*111	SOD

place all spins in the string by the value of the connecting spin (s_{left} or s_{right}), as appropriate.

With all spin values now specified, the number of breaks can be calculated for any of the above algorithms, and the relevant transition probability can be evaluated. The simulated annealing proceeds one step. After that step, all of the previously unknown spin values are again regarded as unknown. (Note: Due to a possible column interchange or block flip, those unknown spins, which previously belonged to particular strings, may now belong to different strings.) The above prescription for replacing unknown spin values by $+1$ or -1 is now repeated, and the transition probabilities are reevaluated. The simulated annealing proceeds another step, etc.

III. APPLICATION

As an example, consider the data presented by Cox *et al.* [3] relating to 14 markers on chromosome 21 that were tested in 99 RH clones. The 14 markers are given in Table I. These are the same data used by Boehnke *et al.* [5] and presented in detail in their Table 1. For our algorithms an entry of 1 in their table becomes $+1$, 0 becomes -1, and a "?" remains unknown and is treated at each step as described above.

All three algorithms were applied to the data in the 99×14 matrix for 200 000 iterations. Initial values of β and incremental steps for increasing β were chosen. This produced a set of permutations with acceptably small values of B, the total number of breaks for a particular configuration. For each run, a ranked set of marker permutations with the smallest values of B was retained.

Table II lists the two distinct "best" orders found by each algorithm in a representative run. As can be seen, the first algorithm, where two nearest-neighbor columns are transposed, does not result in permutations with values of B that are as low as those reached by algorithms 2 and 3. Although in principle, the nearest-neighbor transposition should allow for visiting all possible permutations of the columns, in practice, such exploration is not efficiently accomplished here. The large configuration space, and the empirical nature of selecting β to implement simulated annealing, provide a computational challenge for the first algorithm. The other two algorithms display improved ability to achieve low B values with the chosen set of parameters. Additionally these algorithms reach the same optimal order as that attained by Boehnke *et al.* (see their Table 2), with the same number of breaks. The only difference is that we have retained all 14 markers, whereas they combined markers *S*12 and *S*111, since the latter markers were indistinguishable in the data matrix. Hence for each marker order in their Table 2, we would have two orders.

Although algorithms such as these do not assure that the marker order with the smallest number of breaks is the correct order, inspection of a set of low-lying states leads to some useful information about the stability of clusters of markers that retain their local ordering. For example, consider the set of unique permutations representing the 24 "best" orders obtained from a series of runs of the three algorithms (Table III). We see, e.g., that *S*47 and *SOD* are nearest neighbors in all 24 permutations and appear as the last two markers in 16 permutations. Similarly, the triplet *S*46-*S*4-*S*52 is preserved in 21 permutations and of these, positions 2, 3, and 4 contain *S*46, *S*4, and *S*52, respectively, 15 times. Based on observations such as these, we looked for a general ordering property associated with sets of low-lying configurations.

A. Markovian-like property of low-lying configurations

Let the DNA marker at site j be denoted by f_j, where f_j is a member of the set $\{S16, S48, S46, S4, S52, \ldots\}$. A configuration of sites is denoted by the ordered N-tuple $(f_1, f_2, \ldots, f_N)$. We have used underlining to distinguish a DNA marker such as *S*11 from a spin variable such as s_{11}.

For any configuration of sites, one can define strings $(f_i, f_{i+1}, \ldots, f_{i+m})$ with $i=1,\ldots,N; 0\le m\le N-i$.

Consider a collection $\mathcal{C}$ of distinct configurations. In the collection the probability of a string $(f_i, f_{i+1}, \ldots, f_{i+m})$ is denoted by $P_{i,\ldots,i+m}(f_i, f_{i+1}, \ldots, f_{i+m})$.

Now consider the Markovian-like approximation

TABLE II. Minimum-break orders for representative simulations. See Table I.

Algorithm	Breaks	Marker order													
1	205	APP	*S*8	*S*1	*S*11	*S*16	*S*4	*S*12	*S*18	*S*48	*S*46	*S*52	*S*111	*S*47	SOD
1	207	APP	*S*8	*S*1	*S*11	*S*16	*S*4	*S*12	*S*18	*S*46	*S*48	*S*52	*S*111	*S*47	SOD
2	123	*S*16	*S*48	*S*46	*S*4	*S*52	*S*11	*S*1	*S*18	*S*8	APP	*S*111	*S*12	*S*47	SOD
2	123	*S*16	*S*48	*S*46	*S*4	*S*52	*S*11	*S*1	*S*18	*S*8	APP	*S*12	*S*111	*S*47	SOD
3	123	*S*16	*S*48	*S*46	*S*4	*S*52	*S*11	*S*1	*S*18	*S*8	APP	*S*111	*S*12	*S*47	SOD
3	123	*S*16	*S*48	*S*46	*S*4	*S*52	*S*11	*S*1	*S*18	*S*8	APP	*S*12	*S*111	*S*47	SOD

TABLE III. 24 unique marker orders with relatively small numbers of breaks. See Table I.

Breaks	Marker order													
123	*S16*	*S48*	*S46*	*S4*	*S52*	*S11*	*S1*	*S18*	*S8*	APP	*S111*	*S12*	*S47*	SOD
125	*S16*	*S48*	*S46*	*S4*	*S52*	*S11*	*S1*	*S18*	APP	*S8*	*S12*	*S111*	*S47*	SOD
126	*S48*	*S16*	*S46*	*S4*	*S52*	*S11*	*S1*	*S18*	*S8*	APP	*S12*	*S111*	*S47*	SOD
126	*S48*	*S16*	*S46*	*S4*	*S52*	*S11*	*S1*	*S18*	*S8*	APP	*S111*	*S12*	*S47*	SOD
127	*S11*	*S1*	*S16*	*S48*	*S46*	*S4*	*S52*	*S18*	*S8*	APP	*S111*	*S12*	*S47*	SOD
127	*S16*	*S48*	*S46*	*S4*	*S52*	*S11*	*S1*	*S18*	*S8*	APP	*S12*	*S111*	*S47*	SOD
127	*S46*	*S48*	*S16*	*S4*	*S52*	*S11*	*S1*	*S18*	*S8*	APP	*S12*	*S111*	*S47*	SOD
128	*S11*	*S1*	*S52*	*S4*	*S46*	*S48*	*S16*	*S18*	*S8*	APP	*S12*	*S111*	*S47*	SOD
128	*S11*	*S1*	*S52*	*S4*	*S46*	*S48*	*S16*	*S18*	*S8*	APP	*S111*	*S12*	*S47*	SOD
128	*S16*	*S48*	*S46*	*S4*	*S52*	SOD	*S47*	*S12*	*S111*	APP	*S8*	*S18*	*S1*	*S11*
128	*S16*	*S48*	*S46*	*S4*	*S52*	SOD	*S47*	*S111*	*S12*	APP	*S8*	*S18*	*S1*	*S11*
129	*S11*	*S1*	*S16*	*S48*	*S46*	*S4*	*S52*	*S18*	*S8*	APP	*S12*	*S111*	*S47*	SOD
129	*S11*	*S1*	*S16*	*S48*	*S46*	*S4*	*S52*	*S18*	APP	*S8*	*S111*	*S12*	*S47*	SOD
129	*S16*	*S48*	*S46*	*S4*	*S52*	*S11*	*S1*	*S18*	*S12*	*S111*	*S8*	APP	*S47*	SOD
129	*S16*	*S48*	*S46*	*S4*	*S52*	*S11*	*S1*	*S18*	*S111*	*S12*	*S8*	APP	*S47*	SOD
130	*S16*	*S48*	*S46*	*S4*	*S52*	APP	*S8*	*S12*	*S111*	*S47*	SOD	*S18*	*S1*	*S11*
130	*S16*	*S48*	*S46*	*S4*	*S52*	APP	*S8*	*S111*	*S12*	*S47*	SOD	*S18*	*S1*	*S11*
130	*S16*	*S48*	*S46*	*S4*	*S52*	SOD	*S47*	APP	*S8*	*S12*	*S111*	*S18*	*S1*	*S11*
130	*S16*	*S48*	*S46*	*S4*	*S52*	SOD	*S47*	*S12*	*S111*	*S8*	APP	*S18*	*S1*	*S11*
.130	*S16*	*S48*	*S46*	*S4*	*S52*	SOD	*S47*	*S111*	*S12*	*S8*	APP	*S18*	*S1*	*S11*
130	*S16*	*S48*	*S46*	*S52*	*S4*	SOD	*S47*	*S12*	*S111*	APP	*S8*	*S18*	*S1*	*S11*
130	*S48*	*S16*	*S46*	*S4*	*S52*	*S11*	*S1*	*S18*	*S12*	*S111*	APP	*S8*	*S47*	SOD
130	*S52*	*S4*	*S16*	*S48*	*S46*	*S11*	*S1*	*S18*	*S8*	APP	*S12*	*S111*	*S47*	SOD
130	*S52*	*S4*	*S46*	*S48*	*S16*	*S11*	*S1*	*S18*	*S8*	APP	*S12*	*S111*	*S47*	SOD

$$P_{i,\ldots,i+m}(f_i,f_{i+1},\ldots,f_{i+m})$$

$$\approx P_{i,i+1,i+2}(f_i|f_{i+1},f_{i+2})P_{i+1,i+2,i+3}(f_{i+1}|f_{i+2},f_{i+3})\cdots P_{i+m-2,i+m-1,i+m}(f_{i+m-2}|f_{i+m-1},f_{i+m}) \quad (3.0)$$

$$\times P_{i+m-1,i+m}(f_{i+m-1},f_{i+m}) \quad \text{for } i=1,\ldots,N-2;\ 2\le m\le N-i\ .$$

B. Example 1

Consider the collection of 24 distinct low-lying configurations given in Table III. Let (the number of markers) $N=14$. Look at the cluster ($f_6=F$, $f_7=G$, $f_8=H$, $f_9=I$, $f_{10}=J$) where F denotes the DNA marker *S11*, G denotes *S1*, H denotes *S18*, I denotes *S8*, and J denotes *APP*. This marker assignment corresponds to the ordering of the first row of Table III.

For the above collection we find the frequency

$$P_{6,7,8,9,10}(F,G,H,I,J)=\tfrac{7}{24} \quad (3.1)$$

and we also find the frequencies

$$P_{6,7,8}(F|G,H)=11/11\ , \quad (3.2)$$

$$P_{7,8,9}(G|H,I)=7/11\ , \quad (3.3)$$

$$P_{8,9,10}(H|I,J)=11/11\ , \quad (3.4)$$

$$P_{9,10}(I,J)=11/24\ , \quad (3.5)$$

so the approximation (3.0) with $i=6$, $m=4$ is satisfied by the observed frequencies.

Similarly, for the above configurations $P_{7,8,9,10}(G,H,I,J)=\frac{7}{24}$ and

$$P_{7,8,9,10}(G,H,I,J)\approx P_{7,8,9,10}(G|H,I)P_{8,9,10}(H|I,J)P_{9,10}(I,J)$$
$$=(7/11)(11/11)(11/24)$$
$$=7/24\ . \quad (3.6)$$

C. Example 2

Consider the same collection of 24 distinct low-lying configurations used in example 1. Look at the cluster ($f_2=B$, $f_3=C$, $f_4=D$, $f_5=E$) where B denotes the DNA marker *S48*, C denotes *S46*, D denotes *S4*, E denotes *S52*.

For the above collection we find the frequency

$$P_{2,3,4,5}(B,C,D,E)=12/24\ , \quad (3.7)$$

and we also find the frequencies

$$P_{2,3,4}(B|C,D)=12/15\ , \quad (3.8)$$

$$P_{3,4,5}(C|D,E)=15/16\ , \quad (3.9)$$

$$P_{4,5}(D,E)=16/24\ , \quad (3.10)$$

and again, the approximation (3.0) is satisfied by the observed frequencies.

However, since

$$P_{2,3}(B|C)=13/17\ , \tag{3.11}$$

$$P_{3,4}(C|D)=15/18\ , \tag{3.12}$$

$$P_{4,5}(D|E)=16/16\ , \tag{3.13}$$

$$P_{2,3,4,5}(B,C,D,E)=12/24\ , \tag{3.14}$$

the frequencies do *not* satisfy the standard Markovian approximation

$$P_{2,3,4,5}(B,C,D,E)\neq P_{2,3}(B|C)P_{3,4}(C|D)P_{4,5}(D|E)P_5(E)\ ,$$

$$12/24\neq(13/17)(15/18)(16/16)(16/24)\ , \tag{3.15}$$

$$0.50\neq 0.425\ .$$

In the context of Markov random fields (Spitzer [13]) one could perhaps find a rigorous basis for the observed Markov-like property.

IV. DISCUSSION

The implementation of experimental techniques using RH data provides a bridge between chromosome mapping data generated from families and physical mapping data. RH experiments allow for the relative ordering of genetic markers that are too closely spaced to be detected by family linkage analysis. Additionally, it is not necessary to have polymorphic markers in order to generate useful information. Although not providing the level of resolution of physical mapping, RH mapping can complement and confirm results generated by pulse field gel electrophoresis.

Boehnke *et al.* [5] have presented a full discussion of the advantages and disadvantages of parametric and nonparametric ordering algorithms. As they point out, algorithms that search for minimum break configurations do not allow for estimates of distances between markers, nor do they provide relative likelihoods for one marker order over another. However, as the present work and the work of Boehnke *et al.* demonstrate, retention and inspection of sets of low-lying configurations yield important insight relating to the correlations of markers.

In our study we were interested in learning what properties might be present in a set of configurations with relatively few breaks. It became obvious that the set of low-lying configurations showed the clustering of particular markers. That was reassuring, since in complex optimization problems of the travelling salesman variety, one typically ends up with a set of low-lying configurations and never discovers the true absolute minimum.

One striking feature which we discovered here is that the set of low-lying configurations is described by a Markov-like probability distribution. That distribution contains all of the observed clustering of the markers. If one enlarges the set of low-lying configurations to include configurations with larger and larger numbers of breaks, the validity of the Markov-like approximation deteriorates. We are not in a position to judge whether the relation (3.0) is necessarily a deep or broad result, but we regard it as interesting.

ACKNOWLEDGMENTS

We would like to thank Lynh Wu for her programming help in this project. This research was supported in part by Grant No. GM29177 from the National Institutes of Health (CTF).

[1] S. J. Goss and H. Harris, Nature **255**, 680 (1975).
[2] S. J. Goss and H. Harris, J. Cell Sci. **25**, 39 (1977).
[3] D. R. Cox, M. Burmeister, E. R. Price, S. Kim, and R. M. Myers, Science **250**, 245 (1990).
[4] M. Burmeister, S. Kim, E. R. Price, T. de Lange, U. Tantravahi, R. M. Myers, and D. R. Cox, Genomics **9**, 19 (1990).
[5] M. Boehnke, K. Lange, and D. R. Cox, Am. J. Hum. Genet. **49**, 1174 (1991).
[6] C. T. Falk, Genomics **9**, 120 (1991).
[7] D. T. Bishop and G. P. Crockford, in Genetic Analysis Workshop 7: Issues in Gene Mapping and Detection of Major Genes, edited by J. W. MacCluer, A. Chakravarti, D. R. Cox, D. T. Bishop, S. J. Bale, and M. H. Skolnick [Cytogenet. Cell Genet. **59**, 93 (1992)].
[8] D. E. Weeks, T. Lehner, and J. Ott, in Genetic Analysis Workshop 7: Issues in Gene Mapping and Detection of Major Genes, edited by J. W. MacCluer, A. Chakravarti, D. R. Cox, D. T. Bishop, S. J. Bale, and M. H. Skolnick [Cytogenet. Cell Genet. **59**, 125 (1992)].
[9] H. Falk, Physica **104A**, 459 (1980).
[10] T. M. Liggett, *Interacting Particle Systems* (Springer-Verlag, New York, 1985).
[11] S. Kirkpatrick, C. D. Gelatt, and M. P. Vecchi, Science **220**, 671 (1983).
[12] W. H. Press, B. P. Flannery, S. A. Tuekolsky, and W. T. Vetterling, *Numerical Recipies: The Art of Scientific Computing* (Cambridge University Press, London, 1986), pp. 326–334.
[13] F. Spitzer, *Random Fields and Interacting Particle Systems* (Mathematical Association of America, Oberlin, OH, 1971); Am. Math. Mon. **78**, 142 (1971).